Linear-use CPS translations in the enriched effect calculus

Logical Methods in Computer Science
Vol. 8(4:2)2012, pp. 1–27
www.lmcs-online.org

Submitted Jan. 31, 2012
Published Oct. 5, 2012

LINEAR-USE CPS TRANSLATIONS IN THE ENRICHED EFFECT CALCULUS *

JEFF EGGER [a], RASMUS EJLERS MØGELBERG [b], AND ALEX SIMPSON [c]

[a] Department of Physics and Atmospheric Science, Dalhousie University, Halifax, N.S., Canada
e-mail address: jeffegger@yahoo.ca

[b] IT University of Copenhagen, Copenhagen, Denmark
e-mail address: mogel@itu.dk

[c] LFCS, School of Informatics, University of Edinburgh, Scotland, UK
e-mail address: Alex.Simpson@ed.ac.uk

Abstract. The *enriched effect calculus* (EEC) is an extension of Moggi's computational metalanguage with a selection of primitives from linear logic. This paper explores the enriched effect calculus as a target language for continuation-passing-style (CPS) translations in which the typing of the translations enforces the linear usage of continuations. We first observe that established call-by-value and call-by name linear-use CPS translations of simply-typed lambda-calculus into intuitionistic linear logic (ILL) land in the fragment of ILL given by EEC. These two translations are uniformly generalised by a single generic translation of the enriched effect calculus into itself. As our main theorem, we prove that the generic self-translation of EEC is involutive up to isomorphism. As corollaries, we obtain full completeness results, both for the generic translation, and for the original call-by-value and call-by-name translations.

1. Introduction

Under a continuation-passing-style (CPS) interpretation, a call-by-value program from X to Y is interpreted as a "continuation transformer", that is, as a map $(Y \to R) \to (X \to R)$, where R represents the possible "results" of a computation. Such maps are in one-to-one correspondence with Kleisi maps for the *continuations monad* $((-) \to R) \to R$, introduced by Moggi in [Mog89, Mog91]. In [BORT02], Berdine *et al.* observe that, in many programming situations, continuation transformers satisfy an additional property: their argument, the continuation $Y \to R$, is used just once, that is, it is used *linearly*. Thus a call-by-value program can be more informatively modelled as a linear function

1998 ACM Subject Classification: D3.1, F3.3, F4.1.
Key words and phrases: Continuations, Linear logic, Computational effects.

* The results in this paper first appeared in the proceedings of FoSSaCS 2010, [EMS10].

[a] Research carried out while Egger was at LFCS, University of Edinburgh. Research supported by EPSRC Research Grant "Linear Observations and Computational Effects", and by the Danish Agency for Science, Technology and Innovation.

DOI:10.2168/LMCS-8(4:2)2012

$(Y \to R) \multimap (X \to R)$, corresponding to a Kleisli map for the *linearly-used continuations monad* $((-) \to R) \multimap R$.

One goal of the present paper is to address the question: what is the natural type-theoretic context for modelling linearly-used continuations? With the presence of both intuitionistic ($\to$) and linear ($\multimap$) arrows, *intuitionistic linear logic (ILL)* [Gir87] seems a natural answer. Indeed, ILL has been used as the basis of a systematic study of linearly-used continuations by Hasegawa. In [Has02], he presents a continuation passing style (CPS) translation of Moggi's call-by-value computational λ-calculus into ILL, using the linearly-used continuations monad, and establishes a full completeness result for this. A follow-up paper [Has04] considers call-by-name.

In this paper we use a more general type theory, the *enriched effect calculus (EEC)* introduced in [EMS09, EMS12], as a target language for linear-use CPS translations. On the one hand, EEC can be seen as a fragment of ILL and, as such, its models strictly generalise models of ILL. On the other hand, it is a conservative extension of the standard calculi for modelling computational effects (Moggi's *computational metalanguage* [Mog91], and Levy's *call-by-push-value (CBPV)* [Lev04]) with a selection of constructs from linear logic. In fact, any *adjunction model* of CBPV [Lev05] (and hence any model of Moggi's computational metalanguage) expands to a model of EEC [EMS09, EMS1x]. This provides an abundant supply of computationally interesting models of EEC that are not models of ILL.

The paper begins with a brief presentation of the enriched effect calculus, in Section 2. The standard call-by-value and call-by-name translations of typed λ-calculus into effect calculi (cf. Moggi [Mog91], Filinski [Fil96], Levy [Lev04]) are then reviewed in Section 3, using EEC as the target language. This is followed, in Section 4, by giving corresponding linear-use CPS translations within EEC. The starting point is the observation that Hasegawa's call-by-value [Has02] and call-by-name [Has04] linear-use CPS translations of simply-typed λ-calculus both fall inside the fragment of ILL corresponding to EEC. One contribution of the paper is to show that, using EEC, we can recover these translations in a particularly interesting way. This is achieved by identifying, in Section 5, a single generic linear-use CPS-translation of the entire enriched effect calculus into itself. In Section 6, it is shown how Hasegawa's call-by-value and call-by-name translations are derived from this by composing the generic translation with the standard call-by-value and call-by-name encodings of typed λ-calculus into effect calculi, reviewed in Section 3.

The generic linear-use CPS-translation of EEC into itself is the principal contribution of the paper. It possesses a remarkable property, unexpected in the context of CPS translations: it is involutive up to isomorphism. That is, the translation of a translated term equals the original term modulo type isomorphism. This property is stated as Theorem 5.4, which is the main theorem of the paper. As consequences, we obtain full-completeness results, both for the generic self-translation itself (Theorem 5.5), and also for the call-by-value and call-by-name linear-use CPS translations into EEC, mirroring Hasegawa's results for the translations into ILL.

In the conference presentation of these results [EMS10], the main syntactic theorem was given a semantic proof using category-theoretic models of EEC. In contrast, in the present paper, we provide purely syntactic proofs of all results. It is hoped that this decision will enlarge the potential readership of the paper. Nevertheless, in Section 7, we briefly outline the semantic context within which the syntactic results can be understood. Even at an informal level, the semantic picture provides an illuminating perspective on the definition

and properties of the generic self-translation of EEC. A full treatment of the semantic side, which requires considerable technical machinery, will be presented in a companion paper [EMS1x], devoted entirely to the category-theoretic model theory of EEC.

A few words on the style of the paper. Since the presentation is syntactic, there are many proofs by induction. Some of these have numerous cases. (The proof of Theorem 5.4, for example, has 41 cases.) In order to keep the paper concise and readable, in such proofs, we present only a few illustrative cases, including the most interesting. However, we take care to establish all the side results (for example, the substitution property of Proposition 5.1) needed to make completing the main proofs routine in principle (if lengthy in practice).

2. The enriched effect calculus

The *enriched effect calculus (EEC)* [EMS09, EMS12] is an extension of Moggi's computational metalanguage [Mog91] with constructors from linear type theory. Similar to Filinski's effect PCF [Fil96] and Levy's CBPV [Lev04], it has two notions of types: *value types* and *computation types*. We use $\alpha, \beta, \ldots$ to range over a set of *value type constants*, and $\underline{\alpha}, \underline{\beta}, \ldots$ to range over a disjoint set of *computation type constants*. We then use $\mathsf{A}, \mathsf{B}, \ldots$ to range over *value types*, and $\underline{\mathsf{A}}, \underline{\mathsf{B}}, \ldots$ to range over *computation types*, which are specified by the grammar below.

$$\mathsf{A} ::= \alpha \mid 1 \mid \mathsf{A} \times \mathsf{B} \mid \mathsf{A} \to \mathsf{B} \mid \underline{\mathsf{A}} \mid \underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$$
$$\underline{\mathsf{A}} ::= \underline{\alpha} \mid \underline{1} \mid \underline{\mathsf{A}} \,\&\, \underline{\mathsf{B}} \mid \mathsf{A} \Rightarrow \underline{\mathsf{B}} \mid \underline{\mathsf{I}} \mid\, !\mathsf{A} \mid !\mathsf{A} \otimes \underline{\mathsf{B}} \mid \underline{0} \mid \underline{\mathsf{A}} \oplus \underline{\mathsf{B}} \ .$$

As in [EMS09, EMS12], our notation has been heavily influenced by linear logic. Indeed, EEC can be roughly understood as a fragment of intuitionistic linear logic. However, there are some discrepancies, both in content and in syntax. An important difference is that, in EEC, computation types are the sole source of linearity. Thus linear function space $\underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$ is defined between computation types only. However, the type $\underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$ itself is a value type not a computation type. As discused in *op. cit.*, this choice seems essential for EEC to be compatible with arbitrary (possibly non-commutative) computational effects. A consequence is that the linear function space cannot be iterated (neither $(\underline{\mathsf{A}} \multimap \underline{\mathsf{B}}) \multimap \underline{\mathsf{C}}$ nor $\underline{\mathsf{A}} \multimap (\underline{\mathsf{B}} \multimap \underline{\mathsf{C}})$ is allowed).

Concerning notation, we remark that the type $!\mathsf{A} \otimes \underline{\mathsf{B}}$ is obtained by the application of a single primitive binary type constructor $!(-) \otimes (-)$ to a value type A and computation type $\underline{\mathsf{B}}$. The hybrid notation for this constructor is chosen to emphasise the connection with linear logic. In the present paper, we distinguish notationally between products of computation types $\underline{1}$ and $\underline{\mathsf{A}} \,\&\, \underline{\mathsf{B}}$, and products of value types 1 and $\mathsf{A} \times \mathsf{B}$. Similarly, we distinguish notationally between computation-type function types $\mathsf{A} \Rightarrow \underline{\mathsf{B}}$ (note that the the domain is a value type) and value-type function types $\mathsf{A} \to \mathsf{B}$. These choices, while adding redundancy to the streamlined syntax of [EMS09, EMS12], have the advantage of simplifying certain properties of the syntactic translations we shall give in Section 4. A further redundancy, introduced to simplify the presentation in Section 5, is that we introduce a primitive computation type $\underline{\mathsf{I}}$, which plays a role analogous to the tensor-product unit in linear logic.[1] This is redundant because $\underline{\mathsf{I}}$ can be defined as $!1$. As in linear logic, in addition to the linear isomorphism $\underline{\mathsf{I}} \cong !1$, the type $\underline{\mathsf{I}}$ enjoys the further isomorphisms $!\mathsf{A} \cong !\mathsf{A} \otimes \underline{\mathsf{I}}$, and $\underline{\mathsf{A}} \cong \underline{\mathsf{I}} \multimap \underline{\mathsf{A}}$ in EEC (the latter isomorphism is not linear, since $\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}$ is not a computation

[1] Our choice of notation for units differs from that of linear logic. In linear logic, the tensor unit, which we call $\underline{\mathsf{I}}$, is written 1, and the unit of the linear product &, which we call $\underline{1}$, is written $\top$.

type). Finally, in EEC, the exponential type $!\mathsf{A}$ plays the role of Moggi's monadic type $T\mathsf{A}$ and Levy's type $F\mathsf{A}$. The linear exponential notation is motivated by the many formal analogies between the properties of $!(-)$ in EEC and in ILL. For example, EEC has the type isomorphisms $\mathsf{A} \Rightarrow \underline{\mathsf{B}} \cong !\mathsf{A} \multimap \underline{\mathsf{B}} \cong \mathsf{A} \to \underline{\mathsf{B}}$ (although only the first is a computation type). As in [EMS09, EMS12], we choose to make Levy's U type constructor (see [Lev04]) invisible by including computation types as value types.

The enriched effect calculus has two typing judgements:

$$\text{(i)}\ \ \Gamma \mid - \vdash t : \mathsf{B} \qquad\qquad \text{(ii)}\ \ \Gamma \mid z : \underline{\mathsf{A}} \vdash t : \underline{\mathsf{B}}\ ,$$

where Γ is a context of value-type assignments to variables. On the right of Γ is a *stoup*, which may either be empty, as in the case of judgement (i), or may consist of a unique type assignment $z : \underline{\mathsf{A}}$, in which case the type on the right of the turnstyle is also required to be a computation type, as in (ii). The typing rules are given in Figure 1. In them, Δ ranges over an arbitrary (possibly empty) stoup, and the rules are only applicable in the case of typing judgements that conform to (i) or (ii) above.

Proposition 2.1 (Weakening)**.** *If* $\Gamma \mid \Delta \vdash t : \underline{\mathsf{A}}$ *and variable* x *is not contained in* Γ, Δ *then* $\Gamma, x : \underline{\mathsf{B}} \mid \Delta \vdash t : \underline{\mathsf{A}}$.

Proposition 2.2 (Substitution)**.**
(1) *If* $\Gamma, x : \mathsf{A} \mid \Delta \vdash t : \mathsf{B}$ *and* $\Gamma \mid - \vdash u : \mathsf{A}$ *and then* $\Gamma \mid \Delta \vdash t[u/x] : \mathsf{B}$.
(2) *If* $\Gamma \mid x : \underline{\mathsf{A}} \vdash t : \underline{\mathsf{B}}$ *and* $\Gamma \mid \Delta \vdash u : \underline{\mathsf{A}}$ *then then* $\Gamma \mid \Delta \vdash t[u/x] : \underline{\mathsf{B}}$.

A simple consequence of the propositions above is that EEC satisfies the "shift" property: if $\Gamma \mid x : \underline{\mathsf{A}} \vdash t : \underline{\mathsf{B}}$ then $\Gamma, x : \underline{\mathsf{A}} \mid - \vdash t : \underline{\mathsf{B}}$. See [EMS12] for further discussion of syntactic properties of EEC.

Rules for equalities between typed terms are presented in Figure 2. They are to be considered in addition to the expected (typed) congruence and α-equivalence rules. The equations of Figure 2 have been formulated in such a way that the smallest α-equivalence-respecting congruence containing these equalities is automatically closed under the substitution operations of Proposition 2.2.

The relationship between the enriched effect calculus and other calculi is discussed in detail in [EMS12]. We summarise the main points relevant to the present paper.

The fragment of EEC obtained by removing the type constructors $\mathsf{A} \to \mathsf{B}$, $\underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$, $!\mathsf{A} \otimes \underline{\mathsf{B}}$, $\underline{0}$ and $\underline{\mathsf{A}} \oplus \underline{\mathsf{B}}$ is called the *effect calculus* (EC) in [EMS12].[2] The effect calculus is equivalent to Levy's CBPV (with complex stacks, finitary syntax version) modulo the difference that CBPV has one further type constructor: value-type sums. Since, on the one hand, value-type sums can be easily added to the effect calculus [EMS09], and, on the other, just as easily removed from CBPV, we consider this difference as minor. Thus it seems fair to view the effect calculus (where value-type sums can be included if desired) as, essentially, a reformulation of CBPV using a syntax and presentation influenced by linear logic. In particular, the style of typing rule we have given owes a debt to Barber and Plotkin's Dual Intuitionistic Linear Logic [Bar97]. The influence of linear logic is, of course, even more apparent in the case of the enriched effect calculus. In [EMS09, EMS1x], it is shown that EEC is a conservative extension of EC, thus the presence of the additional linear primitives does not alter the properties of the core type constructors from EC.

[2]This differs mildly from the "effect calculus" of [EMS12] through not having value-type function spaces.

$$\frac{}{\Gamma,\, x\colon \mathsf{A} \mid - \vdash x\colon \mathsf{A}} \qquad \frac{}{\Gamma \mid - \vdash *\colon 1}$$

$$\frac{\Gamma \mid - \vdash t\colon \mathsf{A} \quad \Gamma \mid - \vdash u\colon \mathsf{B}}{\Gamma \mid - \vdash \langle t, u\rangle\colon \mathsf{A}\times\mathsf{B}} \qquad \frac{\Gamma \mid - \vdash t\colon \mathsf{A}\times\mathsf{B}}{\Gamma \mid - \vdash \mathrm{fst}(t)\colon \mathsf{A}} \qquad \frac{\Gamma \mid - \vdash t\colon \mathsf{A}\times\mathsf{B}}{\Gamma \mid - \vdash \mathrm{snd}(t)\colon \mathsf{B}}$$

$$\frac{\Gamma,\, x\colon \mathsf{A} \mid - \vdash t\colon \mathsf{B}}{\Gamma \mid - \vdash \lambda x\colon \mathsf{A}.\, t\colon \mathsf{A}\to\mathsf{B}} \qquad \frac{\Gamma \mid - \vdash s\colon \mathsf{A}\to\mathsf{B} \quad \Gamma \mid - \vdash t\colon \mathsf{A}}{\Gamma \mid - \vdash s(t)\colon \mathsf{B}}$$

$$\frac{}{\Gamma \mid z\colon \underline{\mathsf{A}} \vdash z\colon \underline{\mathsf{A}}} \qquad \frac{}{\Gamma \mid \Delta \vdash \underline{*}\colon \underline{1}}$$

$$\frac{\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}} \quad \Gamma \mid \Delta \vdash u\colon \underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash \underline{\langle t, u\rangle}\colon \underline{\mathsf{A}}\,\&\,\underline{\mathsf{B}}} \qquad \frac{\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}}\,\&\,\underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash \underline{\mathrm{fst}}(t)\colon \underline{\mathsf{A}}} \qquad \frac{\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}}\,\&\,\underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash \underline{\mathrm{snd}}(t)\colon \underline{\mathsf{B}}}$$

$$\frac{\Gamma,\, x\colon \mathsf{A} \mid \Delta \vdash t\colon \underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash \underline{\lambda} x\colon \mathsf{A}.\, t\colon \mathsf{A}\Rightarrow\underline{\mathsf{B}}} \qquad \frac{\Gamma \mid \Delta \vdash s\colon \mathsf{A}\Rightarrow\underline{\mathsf{B}} \quad \Gamma \mid - \vdash t\colon \mathsf{A}}{\Gamma \mid \Delta \vdash s\underline{(t)}\colon \underline{\mathsf{B}}}$$

$$\frac{}{\Gamma \mid - \vdash \top\colon \underline{\mathsf{I}}} \qquad \frac{\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{I}} \quad \Gamma \mid - \vdash u\colon \underline{\mathsf{A}}}{\Gamma \mid \Delta \vdash \mathrm{let}\ \top\ \mathrm{be}\ t\ \mathrm{in}\ u\colon \underline{\mathsf{A}}}$$

$$\frac{\Gamma \mid - \vdash t\colon \mathsf{A}}{\Gamma \mid - \vdash\ !t\colon\ !\mathsf{A}} \qquad \frac{\Gamma \mid \Delta \vdash t\colon\ !\mathsf{A} \quad \Gamma,\, x\colon \mathsf{A} \mid - \vdash u\colon \underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash \mathrm{let}\ !x\ \mathrm{be}\ t\ \mathrm{in}\ u\colon \underline{\mathsf{B}}}$$

$$\frac{\Gamma \mid - \vdash t\colon \mathsf{A} \quad \Gamma \mid \Delta \vdash u\colon \underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash\ !t\otimes u\colon\ !\mathsf{A}\otimes\underline{\mathsf{B}}} \qquad \frac{\Gamma \mid \Delta \vdash s\colon\ !\mathsf{A}\otimes\underline{\mathsf{B}} \quad \Gamma,\, x\colon \mathsf{A} \mid y\colon \underline{\mathsf{B}} \vdash t\colon \underline{\mathsf{C}}}{\Gamma \mid \Delta \vdash \mathrm{let}\ !x\otimes y\ \mathrm{be}\ s\ \mathrm{in}\ t\colon \underline{\mathsf{C}}}$$

$$\frac{\Gamma \mid \Delta \vdash t\colon \underline{0}}{\Gamma \mid \Delta \vdash \underline{?}(t)\colon \underline{\mathsf{A}}} \qquad \frac{\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}}}{\Gamma \mid \Delta \vdash \underline{\mathrm{inl}}(t)\colon \underline{\mathsf{A}}\oplus\underline{\mathsf{B}}} \qquad \frac{\Gamma \mid \Delta \vdash t\colon \underline{\mathsf{B}}}{\Gamma \mid \Delta \vdash \underline{\mathrm{inr}}(t)\colon \underline{\mathsf{A}}\oplus\underline{\mathsf{B}}}$$

$$\frac{\Gamma \mid \Delta \vdash s\colon \underline{\mathsf{A}}\oplus\underline{\mathsf{B}} \quad \Gamma \mid x\colon \underline{\mathsf{A}} \vdash t\colon \underline{\mathsf{C}} \quad \Gamma \mid y\colon \underline{\mathsf{B}} \vdash u\colon \underline{\mathsf{C}}}{\Gamma \mid \Delta \vdash \underline{\mathrm{case}}\, s\, \mathrm{of}\, (\underline{\mathrm{inl}}(x).\, t;\, \underline{\mathrm{inr}}(y).\, u)\colon \underline{\mathsf{C}}}$$

$$\frac{\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t\colon \underline{\mathsf{B}}}{\Gamma \mid - \vdash \lambda^{\circ} z\colon \underline{\mathsf{A}}.\, t\colon \underline{\mathsf{A}}\multimap\underline{\mathsf{B}}} \qquad \frac{\Gamma \mid - \vdash s\colon \underline{\mathsf{A}}\multimap\underline{\mathsf{B}} \quad \Gamma \mid \Delta \vdash t\colon \underline{\mathsf{A}}}{\Gamma \mid \Delta \vdash s[t]\colon \underline{\mathsf{B}}}$$

Figure 1: Typing rules for the enriched effect calculus

$\Gamma \mid - \vdash t = * \colon 1$	if $\Gamma \mid - \vdash t \colon 1$
$\Gamma \mid - \vdash \mathrm{fst}(\langle t, u\rangle) = t \colon \mathsf{A}$	if $\Gamma \mid - \vdash t \colon \mathsf{A}$ and $\Gamma \mid - \vdash u \colon \mathsf{B}$
$\Gamma \mid - \vdash \mathrm{snd}(\langle t, u\rangle) = u \colon \mathsf{B}$	if $\Gamma \mid - \vdash t \colon \mathsf{A}$ and $\Gamma \mid - \vdash u \colon \mathsf{B}$
$\Gamma \mid - \vdash \langle \mathrm{fst}(t), \mathrm{snd}(t)\rangle = t \colon \mathsf{A} \times \mathsf{B}$	if $\Gamma \mid - \vdash t \colon \mathsf{A} \times \mathsf{B}$
$\Gamma \mid - \vdash (\lambda x \colon \mathsf{A}.\, t)(u) = t[u/x] \colon \mathsf{B}$	if $\Gamma, x \colon \mathsf{A} \mid - \vdash t \colon \mathsf{B}$ and $\Gamma \mid - \vdash u \colon \mathsf{A}$
$\Gamma \mid - \vdash \lambda x \colon \mathsf{A}.\, (t(x)) = t \colon \mathsf{A} \to \mathsf{B}$	if $\Gamma \mid - \vdash t \colon \mathsf{A} \to \mathsf{B}$ and $x \notin \Gamma$
$\Gamma \mid \Delta \vdash t = \underline{*} \colon \underline{1}$	if $\Gamma \mid \Delta \vdash t \colon \underline{1}$
$\Gamma \mid \Delta \vdash \underline{\mathrm{fst}}(\underline{\langle} t, u \underline{\rangle}) = t \colon \underline{\mathsf{A}}$	if $\Gamma \mid \Delta \vdash t \colon \underline{\mathsf{A}}$ and $\Gamma \mid \Delta \vdash u \colon \underline{\mathsf{B}}$
$\Gamma \mid \Delta \vdash \underline{\mathrm{snd}}(\underline{\langle} t, u \underline{\rangle}) = u \colon \underline{\mathsf{B}}$	if $\Gamma \mid \Delta \vdash t \colon \underline{\mathsf{A}}$ and $\Gamma \mid \Delta \vdash u \colon \underline{\mathsf{B}}$
$\Gamma \mid \Delta \vdash \underline{\langle} \underline{\mathrm{fst}}(t), \underline{\mathrm{snd}}(t) \underline{\rangle} = t \colon \underline{\mathsf{A}} \,\&\, \underline{\mathsf{B}}$	if $\Gamma \mid \Delta \vdash t \colon \underline{\mathsf{A}} \,\&\, \underline{\mathsf{B}}$
$\Gamma \mid \Delta \vdash (\underline{\lambda} x \colon \mathsf{A}.\, t)(\underline{u}) = t[u/x] \colon \underline{\mathsf{B}}$	if $\Gamma, x \colon \mathsf{A} \mid \Delta \vdash t \colon \underline{\mathsf{B}}$ and $\Gamma \mid - \vdash u \colon \mathsf{A}$
$\Gamma \mid \Delta \vdash \underline{\lambda} x \colon \mathsf{A}.\, (t(\underline{x})) = t \colon \mathsf{A} \Rightarrow \underline{\mathsf{B}}$	if $\Gamma \mid \Delta \vdash t \colon \mathsf{A} \Rightarrow \underline{\mathsf{B}}$ and $x \notin \Gamma, \Delta$
$\Gamma \mid - \vdash \mathrm{let}\ \top \mathrm{be}\ \top \mathrm{in}\ t = t \colon \underline{\mathsf{A}}$	if $\Gamma \mid - \vdash t \colon \underline{\mathsf{A}}$
$\Gamma \mid \Delta \vdash \mathrm{let}\ \top \mathrm{be}\ t \mathrm{in}\ u[\top/x] = u[t/x] \colon \underline{\mathsf{A}}$	if $\Gamma \mid \Delta \vdash t \colon \underline{\mathsf{I}}$ and $\Gamma \mid x \colon \underline{\mathsf{I}} \vdash u \colon \underline{\mathsf{A}}$
$\Gamma \mid - \vdash \mathrm{let}\ !x \mathrm{be}\ !t \mathrm{in}\ u = u[t/x] \colon \underline{\mathsf{B}}$	if $\Gamma \mid - \vdash t \colon \mathsf{A}$ and $\Gamma, x \colon \mathsf{A} \mid - \vdash u \colon \underline{\mathsf{B}}$
$\Gamma \mid \Delta \vdash \mathrm{let}\ !x \mathrm{be}\ t \mathrm{in}\ u[!x/y] = u[t/y] \colon \underline{\mathsf{B}}$	if $\Gamma \mid \Delta \vdash t \colon !\mathsf{A}$ and $\Gamma \mid y \colon !\mathsf{A} \vdash u \colon \underline{\mathsf{B}}$
$\Gamma \mid \Delta \vdash \mathrm{let}\ !x \otimes y \mathrm{be}\ !t \otimes s \mathrm{in}\ u = u[t,s/x,y] \colon \underline{\mathsf{C}}$	if $\Gamma \mid - \vdash t \colon \mathsf{A}$, $\Gamma \mid \Delta \vdash s \colon \underline{\mathsf{B}}$, and $\Gamma, x \colon \mathsf{A} \mid y \colon \underline{\mathsf{B}} \vdash u \colon \underline{\mathsf{C}}$
$\Gamma \mid \Delta \vdash \mathrm{let}\ !x \otimes y \mathrm{be}\ t \mathrm{in}\ u[!x \otimes y/z] = u[t/z] \colon \underline{\mathsf{C}}$	if $\Gamma \mid \Delta \vdash t \colon !\mathsf{A} \otimes \underline{\mathsf{B}}$ and $\Gamma \mid z \colon !\mathsf{A} \otimes \underline{\mathsf{B}} \vdash u \colon \underline{\mathsf{C}}$
$\Gamma \mid \Delta \vdash \underline{?}(t) = u[t/x] \colon \underline{\mathsf{A}}$	if $\Gamma \mid \Delta \vdash t \colon \underline{0}$ and $\Gamma \mid x \colon \underline{0} \vdash u \colon \underline{\mathsf{A}}$
$\Gamma \mid \Delta \vdash \underline{\mathrm{case}}\, \underline{\mathrm{inl}}(t) \,\mathrm{of}\, (\underline{\mathrm{inl}}(x).\, u;\ \underline{\mathrm{inr}}(y).\, u') = u[t/x] : \underline{\mathsf{C}}$	if $\Gamma \mid x \colon \underline{\mathsf{A}} \vdash u \colon \underline{\mathsf{C}}$ and $\Gamma \mid y \colon \underline{\mathsf{B}} \vdash u' \colon \underline{\mathsf{C}}$ and $\Gamma \mid \Delta \vdash t \colon \underline{\mathsf{A}}$
$\Gamma \mid \Delta \vdash \underline{\mathrm{case}}\, \underline{\mathrm{inr}}(t) \,\mathrm{of}\, (\underline{\mathrm{inl}}(x).\, u;\ \underline{\mathrm{inr}}(y).\, u') = u'[t/y] : \underline{\mathsf{C}}$	if $\Gamma \mid x \colon \underline{\mathsf{A}} \vdash u \colon \underline{\mathsf{C}}$ and $\Gamma \mid y \colon \underline{\mathsf{B}} \vdash u' \colon \underline{\mathsf{C}}$ and $\Gamma \mid \Delta \vdash t \colon \underline{\mathsf{B}}$
$\Gamma \mid \Delta \vdash \underline{\mathrm{case}}\, t \,\mathrm{of}\, (\underline{\mathrm{inl}}(x).\, u[\underline{\mathrm{inl}}(x)/z];\ \underline{\mathrm{inr}}(y).\, u[\underline{\mathrm{inr}}(y)/z]) = u[t/z] : \underline{\mathsf{C}}$	if $\Gamma \mid \Delta \vdash t \colon \underline{\mathsf{A}} \oplus \underline{\mathsf{B}}$ and $\Gamma \mid z \colon \underline{\mathsf{A}} \oplus \underline{\mathsf{B}} \vdash u \colon \underline{\mathsf{C}}$
$\Gamma \mid \Delta \vdash (\lambda^{\circ} x \colon \underline{\mathsf{A}}.\, t)[u] = t[u/x] \colon \underline{\mathsf{B}}$	if $\Gamma \mid x \colon \underline{\mathsf{A}} \vdash t \colon \underline{\mathsf{B}}$ and $\Gamma \mid \Delta \vdash u \colon \underline{\mathsf{A}}$
$\Gamma \mid - \vdash \lambda^{\circ} x \colon \underline{\mathsf{A}}.\, (t[x]) = t \colon \underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$	if $\Gamma \mid - \vdash t \colon \underline{\mathsf{A}} \multimap \underline{\mathsf{B}}$ and $x \notin \Gamma$

Figure 2: Equality rules for the enriched effect calculus

$$\frac{}{\Theta,\, x\colon\sigma \vdash x\colon\sigma} \qquad \frac{}{\Theta \mid - \vdash *\colon 1}$$

$$\frac{\Theta \vdash M\colon\sigma \quad \Theta \vdash N\colon\tau}{\Theta \vdash \langle M, N\rangle\colon\sigma\times\tau} \qquad \frac{\Theta \vdash M\colon\sigma\times\tau}{\Theta \vdash \mathrm{fst}(M)\colon\sigma} \qquad \frac{\Theta \vdash M\colon\sigma\times\tau}{\Theta \vdash \mathrm{snd}(M)\colon\tau}$$

$$\frac{\Theta,\, x\colon\sigma \vdash M\colon\tau}{\Theta \vdash \lambda x\colon\sigma.\, M\colon\sigma\to\tau} \qquad \frac{\Theta \vdash M\colon\sigma\to\tau \quad \Theta \vdash N\colon\sigma}{\Theta \vdash MN\colon\tau}$$

Figure 3: Typing rules for simply-typed λ-calculus

It is also natural to compare EEC with ILL. In the present paper, we do this informally and crudely.[3] We include EEC in ILL by ignoring the distinction between value and computation types, and mapping all type constructors to their evident (mainly synonymous) linear counterparts. For example, both $\to$ and $\Rightarrow$ get mapped to the intuitionistic function space of ILL; both $\times$ and & get mapped to the linear "with" &; both 1 and $\underline{1}$ get mapped to the unit of the intuitionistic "with", which is usually denoted $\top$; and $\underline{\mathbf{!}}$ gets mapped to the unit of the linear tensor, which is usually denoted 1. This translation from EEC to ILL is "sound" in the sense that terms that are equal in EEC get mapped to equal terms in ILL. (This is a consequence of the simple observation that the typing rules and equations of EEC are all have direct counterparts in the presentation of ILL of [Bar97].) However, the translation is not "complete": terms of the same type whose translations are equal in ILL need not be equal in EEC. It is also not "full", there exist terms in ILL whose type lies in the EEC fragment of ILL, but which are not equal to the translation of any EEC term.

3. Call-by-value and call-by-name translations into EEC

There is a standard call-by-value translation of typed λ-calculus into Moggi's computational metalanguage [Mog91], Filinski's effect PCF [Fil96], and Levy's CBPV [Lev04]. Similarly, there is a standard call-by-name translation into the latter two, which exploits the existence of computation types.[4] We recall these translations using the syntax of the enriched effect calculus.

As a source calculus, we use the simply-typed λ-calculus with types $\sigma, \tau, \ldots$ given by:

$$\sigma \;::=\; \alpha \mid 1 \mid \sigma\times\tau \mid \sigma\to\tau \;,$$

where α ranges over a collection of type constants. We use Θ to range over finite contexts $x_1\colon\sigma_1, \ldots, x_n\colon\sigma_n$, and M, N to range over terms of the simply-typed λ-calculus, using the syntax given by the typing rules in Figure 3.

The call-by-value interpretation translates a type σ to a value type σ^{v}. The call-by-name interpretation translates it to a computation type σ^{n}. Both translations are defined in Figure 4. For the translations of type constants, we assume that each type constant α of the typed λ-calculus, is included as a value-type constant in EEC, and has an associated computation-type constant $\underline{\alpha}$. Note that the definition of $(\sigma \to \tau)^{\mathrm{v}}$ could equally well

[3] A less crude comparison retains the distinction between computation and value type, and compares with Benton's mixed linear/non-linear logic [Ben95], in which a similar distinction is maintained. Such a comparison produces identical results: the translation is sound, but neither complete nor full.

[4] Moggi [Mog91] and Benton and Wadler [BW96] refer to a different "lazy" translation as call-by-name.

$$\begin{aligned} \alpha^{\mathrm{v}} &= \alpha & \alpha^{\mathrm{n}} &= \underline{\alpha} \\ 1^{\mathrm{v}} &= 1 & 1^{\mathrm{n}} &= \underline{1} \\ (\sigma \times \tau)^{\mathrm{v}} &= \sigma^{\mathrm{v}} \times \tau^{\mathrm{v}} & (\sigma \times \tau)^{\mathrm{n}} &= \sigma^{\mathrm{n}} \,\&\, \tau^{\mathrm{n}} \\ (\sigma \to \tau)^{\mathrm{v}} &= \sigma^{\mathrm{v}} \to !(\tau^{\mathrm{v}}) & (\sigma \to \tau)^{\mathrm{n}} &= \sigma^{\mathrm{n}} \Rightarrow \tau^{\mathrm{n}} \ . \end{aligned}$$

Figure 4: Cbv and cbn translations of simply-typed λ-calculus

have been given as $\sigma^{\mathrm{v}} \Rightarrow !\tau^{\mathrm{v}}$, which, considered as a value type, is isomorphic to the gven translation. Our reason for instead choosing $\sigma^{\mathrm{v}} \to !(\tau^{\mathrm{v}})$ is that this simplifies the statement of Theorem 6.1 below.

On terms, the cbv translation maps a judgement $x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash M : \tau$ to

$$x_1 : \sigma_1^{\mathrm{v}}, \ldots, x_n : \sigma_n^{\mathrm{v}} \mid - \vdash M^{\mathrm{v}} : !\tau^{\mathrm{v}} \ .$$

It is inductively defined by:

$$\begin{aligned} x^{\mathrm{v}} &= !x \\ *^{\mathrm{v}} &= !* \\ \langle M, N \rangle^{\mathrm{v}} &= \text{let } !x \text{ be } M^{\mathrm{v}} \text{ in let } !y \text{ be } N^{\mathrm{v}} \text{ in } !\langle x, y \rangle \\ (\mathrm{fst}(M))^{\mathrm{v}} &= \text{let } !z \text{ be } M^{\mathrm{v}} \text{ in } !\mathrm{fst}(z) \\ (\mathrm{snd}(M))^{\mathrm{v}} &= \text{let } !z \text{ be } M^{\mathrm{v}} \text{ in } !\mathrm{snd}(z) \\ (\lambda x : \sigma.\, M)^{\mathrm{v}} &= !(\lambda x : \sigma^{\mathrm{v}}.\, M^{\mathrm{v}}) \\ (MN)^{\mathrm{v}} &= \text{let } !f \text{ be } M^{\mathrm{v}} \text{ in let } !x \text{ be } N^{\mathrm{v}} \text{ in } f(x) \ . \end{aligned}$$

The cbn translation maps a judgement $x_1 : \sigma_1, \ldots, x_n : \sigma_n \vdash M : \tau$ to

$$x_1 : \sigma_1^{\mathrm{n}}, \ldots, x_n : \sigma_n^{\mathrm{n}} \mid - \vdash M^{\mathrm{n}} : \tau^{\mathrm{n}} \ ,$$

and simply uses the constructs associated with the computation-type constructors $\underline{1}$, $\&$ and $\Rightarrow$ to mimic the corresponding constructs for 1, $\times$ and $\to$ in the simply-typed λ-calculus. Since this is essentially trivial, we omit the details.

The call-by-value and call-by-name translations into EEC induce equational theories on simply-typed λ-terms. In the case of call-by-value, the resulting equational theory is that of Moggi's *computational λ-calculus*, λ_c, [Mog89]. In the case of call-by-name, it is the usual $\beta\eta$-equality theory. The propositions below state this formally, and also assert that the translations into EEC are *full* in the sense that every EEC term of translated type is equal to the translation of a simply-typed term. In the statements, and henceforth, we write $\Theta \vdash M =_{\lambda_c} N : \tau$ for equality in Moggi's λ_c, and $\Theta \vdash M =_{\beta\eta} N : \tau$ for $\beta\eta$-equality.

Proposition 3.1 (Soundness and full completeness of $(\cdot)^{\mathrm{v}}$)**.**
(1) *If* $\Theta \vdash M =_{\lambda_c} N : \tau$ *then* $\Theta^{\mathrm{v}} \mid - \vdash M^{\mathrm{v}} = N^{\mathrm{v}} : !\tau^{\mathrm{v}}$.
(2) *If* $\Theta \vdash M, N : \tau$ *and* $\Theta^{\mathrm{v}} \mid - \vdash M^{\mathrm{v}} = N^{\mathrm{v}} : !\tau^{\mathrm{v}}$ *then* $\Theta \vdash M =_{\lambda_c} N : \tau$.
(3) *If* $\Theta^{\mathrm{v}} \mid - \vdash t : !\tau^{\mathrm{v}}$ *then there exists a term* $\Theta \vdash M : \tau$ *such that* $\Theta^{\mathrm{v}} \mid - \vdash M^{\mathrm{v}} = t : !\tau^{\mathrm{v}}$.

Here, statement 1 asserts soundness, statement 2 completeness, and statement 3 fullness.

Proposition 3.2 (Soundness and full completeness of $(\cdot)^{\mathrm{n}}$)**.**
(1) *If* $\Theta \vdash M =_{\beta\eta} N : \tau$ *then* $\Theta^{\mathrm{n}} \mid - \vdash M^{\mathrm{n}} = N^{\mathrm{n}} : \tau^{\mathrm{n}}$.
(2) *If* $\Theta \vdash M, N : \tau$ *and* $\Theta^{\mathrm{n}} \mid - \vdash M^{\mathrm{n}} = N^{\mathrm{n}} : \tau^{\mathrm{n}}$ *then* $\Theta \vdash M =_{\beta\eta} N : \tau$.

(3) *If* $\Theta^{\mathrm{n}} \mid - \vdash t \colon \tau^{\mathrm{n}}$ *then there exists a term* $\Theta \vdash M \colon \tau$ *such that* $\Theta^{\mathrm{n}} \mid - \vdash M^{\mathrm{n}} = t \colon \tau^{\mathrm{n}}$.

Outline proof of Propositions 3.1 and 3.2. The call-by-value and call-by-name translations of typed λ-calculus into Levy's CBPV are known to be fully complete [Lev04, Appendix A]. These translations thus transfer to the *effect calculus* (EC) of [EMS12], which is essentially equivalent to CBPV. The resulting translations into EC are essentially identical to those given above, modulo the inclusion of EC in the enriched effect calculus. This inclusion is shown to be fully complete in [EMS09, EMS1x]. □

The repeat appearance of the word "essentially" in the outline proof above calls for clarification. As already discussed in Section 2, the equivalence between CBPV and the effect calculus requires choosing the correct version of CBPV (with complex stacks and finitary syntax), and ignoring the fact that CBPV has value-type sums but EC does not. Anyway, such issues are a distraction here, since the translations do not involve sum types, and Levy's proofs of full completeness transfer directly to EC. Second, the call-by-value translation we have given into EEC is not literally identical to the translation into EC. The difference is that, in the case of EC (as defined in [EMS12], see Section 2), one has to define $(\sigma \to \tau)^{\mathrm{v}} = \sigma^{\mathrm{v}} \Rightarrow !\tau^{\mathrm{v}}$, because value-type function space is not available. This difference is, however, trivial since the two function spaces are isomorphic as value types.

Via the inclusion of EEC as a fragment of ILL, the translations defined above can also be viewed as translations into ILL. In the case of call-by-value, the resulting translation into ILL is exactly Benton and Wadler's call-by-value translation from [BW96]. As emphasised in *op. cit.*, this translation is not complete relative to $=_{\lambda_c}$ because it enforces the *commutativity* of effects. For example, the two terms below,

$$f \colon 1 \to 1,\ g \colon 1 \to 1 \vdash (\lambda x \colon 1.\ \lambda y \colon 1.\ *)(f*)(g*) \colon 1 \tag{3.1}$$

$$f \colon 1 \to 1,\ g \colon 1 \to 1 \vdash (\lambda x \colon 1.\ \lambda y \colon 1.\ *)(g*)(f*) \colon 1\ , \tag{3.2}$$

which are not equated by $=_{\lambda_c}$, are equated by the translation. It is also known that the call-by-value translation into ILL is not full [Has02].

4. Linearly-used continuations in EEC

In [Plo75], Plotkin gave continuation passing style (CPS) translations of call-by-value and call-by-name λ-calculi into the λ-calculus. As emerged from the work of Moggi [Mog89, Mog91], the typed version of Plotkin's call-by-value translation is sound relative to the equational theory, $=_{\lambda_c}$, of the computational λ-calculus. Although Plotkin's original call-by-name translation validates only the β-law, a variation due to Reus and Streicher [RS98] is sound for $=_{\beta\eta}$. A feature shared by all these translations is that the usage of continuations within them is linear. This aspect has been formalized by Hasegawa. In [Has02], he studies a call-by-value translation from typed λ-calculus into intuitionistic linear type theory (ILL) in which the types of the translation enforce the linear usage of continuations. In essence, this translation is Plotkin's original call-by-value translation, but carried out within a linear typing discipline. In [Has04], Hasegawa gives a corresponding linear version of the (Reus-Streicher) call-by-name CPS translation. Although Hasegawa's translations are into ILL,

$$\begin{aligned}
\alpha^{\mathrm{v}\underline{\mathsf{R}}} &= \alpha & \alpha^{\mathrm{n}\underline{\mathsf{R}}} &= \underline{\alpha} \\
1^{\mathrm{v}\underline{\mathsf{R}}} &= 1 & 1^{\mathrm{n}\underline{\mathsf{R}}} &= \underline{0} \\
(\sigma \times \tau)^{\mathrm{v}\underline{\mathsf{R}}} &= \sigma^{\mathrm{v}\underline{\mathsf{R}}} \times \tau^{\mathrm{v}\underline{\mathsf{R}}} & (\sigma \times \tau)^{\mathrm{n}\underline{\mathsf{R}}} &= \sigma^{\mathrm{n}\underline{\mathsf{R}}} \oplus \tau^{\mathrm{n}\underline{\mathsf{R}}} \\
(\sigma \to \tau)^{\mathrm{v}\underline{\mathsf{R}}} &= \sigma^{\mathrm{v}\underline{\mathsf{R}}} \to ((\tau^{\mathrm{v}\underline{\mathsf{R}}} \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}) & (\sigma \to \tau)^{\mathrm{n}\underline{\mathsf{R}}} &= !(\sigma^{\mathrm{n}\underline{\mathsf{R}}} \multimap \underline{\mathsf{R}}) \otimes \tau^{\mathrm{n}\underline{\mathsf{R}}} \ .
\end{aligned}$$

Figure 5: Cbv and cbn linear-use CPS translations of typed λ-calculus.

one sees straightforwardly that they land inside the EEC fragment of ILL.[5] We now recall these translations, defining them directly as translations into EEC.

The call-by-value interpretation translates a type σ to a value type $\sigma^{\mathrm{v}\underline{\mathsf{R}}}$, and the call-by-name interpretation translates σ to a computation type $\sigma^{\mathrm{n}\underline{\mathsf{R}}}$, as defined in Figure 5. As is standard for CPS translations, they are defined relative to the choice of a "result" type, $\underline{\mathsf{R}}$. Using EEC as the target language, it is essential that $\underline{\mathsf{R}}$ be a computation type, otherwise the translations would not produce legal types. Unless specified otherwise, we let $\underline{\mathsf{R}}$ be an arbitrary but fixed computation type. However, we shall often need to specify otherwise. As will be seen, many results below will work in two special cases only: when $\underline{\mathsf{R}}$ is either a computation-type constant or the type $\underline{\mathsf{I}}$.

We remark that the combination of function-space constructs that appears in the call-by-value translation of $\sigma \to \tau$, in Figure 5, is forced by the desire to ensure that continuations are linearly used. The linear usage itself is implemented by selecting $\multimap$ for the right-hand arrow. This, in turn, requires the computation-type arrow $\Rightarrow$ to be used in the type $\tau^{\mathrm{v}\underline{\mathsf{R}}} \Rightarrow \underline{\mathsf{R}}$, which types continations. The left-hand arrow is then forced to be $\to$ since its codomain $(\tau^{\mathrm{v}\underline{\mathsf{R}}} \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}$ is a value type. (It is possible to reduce the number of different function-space constructs that appear in the definition of $(\sigma \to \tau)^{\mathrm{v}\underline{\mathsf{R}}}$ to two. For example, one could define $(\sigma \to \tau)^{\mathrm{v}\underline{\mathsf{R}}}$ to be $(\tau^{\mathrm{v}\underline{\mathsf{R}}} \Rightarrow \underline{\mathsf{R}}) \multimap (\sigma^{\mathrm{v}\underline{\mathsf{R}}} \Rightarrow \underline{\mathsf{R}})$, which is isomorphic to the definition of Figure 5. Another possibility is to reformulate EEC using a single type constructor to implement both value-type and computation-type function spaces, as in the conference version of this paper [EMS10]. However, both these alternatives have the disadvantage, compared with the route we have taken, of complicating the results of Sections 5 and 6.

For a typing context $\Theta = x_1 : \sigma_1, \ldots, x_n : \sigma_n$, define

$$\Theta^{\mathrm{v}\underline{\mathsf{R}}} = x_1 : \sigma_1^{\mathrm{v}\underline{\mathsf{R}}}, \ldots, x_n : \sigma_n^{\mathrm{v}\underline{\mathsf{R}}} \ .$$

Then the cbv translation on terms [Has02] maps a judgement $\Theta \vdash M : \tau$ to

$$\Theta^{\mathrm{v}\underline{\mathsf{R}}} \mid - \vdash M^{\mathrm{v}\underline{\mathsf{R}}} : (\tau^{\mathrm{v}\underline{\mathsf{R}}} \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}} \ ,$$

[5] Actually, in [Has04], Hasegawa gives a call-by-name translation for a variant of Parigot's $\lambda\mu$-calculus [Par92] extending typed λ-calculus. The full translation goes outside of EEC. Here, we consider just the translation restricted to typed λ-calculus, which does land in EEC.

It is defined inductively by (using the typings of Figure 3):

$$\begin{aligned}
x^{\mathrm{v}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon \sigma^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathsf{R}}.\ k\underline{(}x\underline{)}\\
^{\mathrm{v}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon 1 \Rightarrow \underline{\mathsf{R}}.\ k\underline{(}\underline{)}\\
\langle M,N\rangle^{\mathrm{v}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon (\sigma^{\mathrm{v}\underline{\mathrm{R}}}\times\tau^{\mathrm{v}\underline{\mathrm{R}}}) \Rightarrow \underline{\mathsf{R}}.\ M^{\mathrm{v}\underline{\mathrm{R}}}[\underline{\lambda}x\colon\sigma^{\mathrm{v}\underline{\mathrm{R}}}.\ N^{\mathrm{v}\underline{\mathrm{R}}}[\underline{\lambda}y\colon\tau^{\mathrm{v}\underline{\mathrm{R}}}.\ k\underline{(}\langle x,y\rangle\underline{)}]]\\
(\mathrm{fst}(M))^{\mathrm{v}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon \sigma^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathsf{R}}.\ M^{\mathrm{v}\underline{\mathrm{R}}}[\underline{\lambda}z\colon\sigma^{\mathrm{v}\underline{\mathrm{R}}}\times\tau^{\mathrm{v}\underline{\mathrm{R}}}.\ k\underline{(}\mathrm{fst}(z)\underline{)}]\\
(\mathrm{snd}(M))^{\mathrm{v}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon \tau^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathsf{R}}.\ M^{\mathrm{v}\underline{\mathrm{R}}}[\underline{\lambda}z\colon\sigma^{\mathrm{v}\underline{\mathrm{R}}}\times\tau^{\mathrm{v}\underline{\mathrm{R}}}.\ k\underline{(}\mathrm{snd}(z)\underline{)}]\\
(\lambda x\colon\sigma.\ M)^{\mathrm{v}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon (\sigma^{\mathrm{v}\underline{\mathrm{R}}}\to(\tau^{\mathrm{v}\underline{\mathrm{R}}}\Rightarrow\underline{\mathsf{R}})\multimap\underline{\mathsf{R}}) \Rightarrow \underline{\mathsf{R}}.\ k\underline{(}\lambda x\colon\sigma^{\mathrm{v}\underline{\mathrm{R}}}\,M^{\mathrm{v}\underline{\mathrm{R}}}.\ \underline{)}\\
(MN)^{\mathrm{v}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon \tau^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathsf{R}}.\ M^{\mathrm{v}\underline{\mathrm{R}}}[\underline{\lambda}f\colon\sigma^{\mathrm{v}\underline{\mathrm{R}}}\to(\tau^{\mathrm{v}\underline{\mathrm{R}}}\Rightarrow\underline{\mathsf{R}})\multimap\underline{\mathsf{R}}.\ N^{\mathrm{v}\underline{\mathrm{R}}}[\underline{\lambda}x\colon\sigma^{\mathrm{v}\underline{\mathrm{R}}}.\ f(x)[k]]]
\end{aligned}$$

Similarly, define

$$\Theta^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}} \;=\; x_1\colon\sigma_1^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}},\ldots,x_n\colon\sigma_n^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}\ .$$

The cbn translation [Has04] maps a typing judgement $\Theta\vdash M\colon\tau$, as above, to

$$\Theta^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}\mid-\ \vdash\ M^{\mathrm{n}\underline{\mathrm{R}}}\colon\tau^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}\ .$$

Its inductive definition is given by:

$$\begin{aligned}
x^{\mathrm{n}\underline{\mathrm{R}}} &= x\\
*^{\mathrm{n}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon\underline{0}.\ \underline{?}(k)\\
\langle M,N\rangle^{\mathrm{n}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon\sigma^{\mathrm{v}\underline{\mathrm{R}}}\oplus\tau^{\mathrm{v}\underline{\mathrm{R}}}.\ \underline{\mathrm{case}}\,k\,\mathrm{of}\,(\underline{\mathrm{inl}}(x).\ M^{\mathrm{n}\underline{\mathrm{R}}}[x];\ \underline{\mathrm{inr}}(\,y).\ N^{\mathrm{n}\underline{\mathrm{R}}}[y])\\
(\mathrm{fst}(M))^{\mathrm{n}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon\sigma^{\mathrm{n}\underline{\mathrm{R}}}.\ M^{\mathrm{n}\underline{\mathrm{R}}}[\underline{\mathrm{inl}}(k)]\\
(\mathrm{snd}(M))^{\mathrm{n}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon\tau^{\mathrm{n}\underline{\mathrm{R}}}.\ M^{\mathrm{n}\underline{\mathrm{R}}}[\underline{\mathrm{inr}}(k)]\\
(\lambda x\colon\sigma.\ M)^{\mathrm{n}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon!(\sigma^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}})\otimes\tau^{\mathrm{n}\underline{\mathrm{R}}}.\ \mathrm{let}\ !x\otimes h\ \mathrm{be}\ k\ \mathrm{in}\ M^{\mathrm{n}\underline{\mathrm{R}}}[h]\\
(MN)^{\mathrm{n}\underline{\mathrm{R}}} &= \lambda^{\circ}k\colon\tau^{\mathrm{n}\underline{\mathrm{R}}}.\ M^{\mathrm{n}\underline{\mathrm{R}}}[!(N^{\mathrm{n}\underline{\mathrm{R}}})\otimes k]
\end{aligned}$$

The results below list the properties we shall establish of the two translations. Proofs will be given in Section 6.

Proposition 4.1 (Soundness of $(\cdot)^{\mathrm{v}\underline{\mathrm{R}}}$)**.** *If* $\Theta\ \vdash\ M=_{\lambda_c}N\colon\tau$ *then* $\Theta^{\mathrm{v}\underline{\mathrm{R}}}\mid-\ \vdash\ M^{\mathrm{v}\underline{\mathrm{R}}}=N^{\mathrm{v}\underline{\mathrm{R}}}\colon(\tau^{\mathrm{v}\underline{\mathrm{R}}}\Rightarrow\underline{\mathsf{R}})\multimap\underline{\mathsf{R}}$.

Proposition 4.2 (Soundness of $(\cdot)^{\mathrm{n}\underline{\mathrm{R}}}$)**.** *If* $\Theta\ \vdash\ M=_{\beta\eta}N\colon\tau$ *then* $\Theta^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}\mid-\ \vdash\ M^{\mathrm{n}\underline{\mathrm{R}}}=N^{\mathrm{n}\underline{\mathrm{R}}}\colon\tau^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}$.

Theorem 4.3 (Full completeness of $(\cdot)^{\mathrm{v}\underline{\mathrm{R}}}$)**.** *Suppose* $\underline{\mathsf{R}}$ *is either: (i) a computation-type constant, or (ii) the type* $\underline{\mathsf{I}}$*. Then:*

(1) *If* $\Theta\vdash M,N\colon\tau$ *and* $\Theta^{\mathrm{v}\underline{\mathrm{R}}}\mid-\ \vdash\ M^{\mathrm{v}\underline{\mathrm{R}}}=N^{\mathrm{v}\underline{\mathrm{R}}}\colon(\tau^{\mathrm{v}\underline{\mathrm{R}}}\Rightarrow\underline{\mathsf{R}})\multimap\underline{\mathsf{R}}$ *then* $\Theta\ \vdash\ M=_{\lambda_c}N\colon\tau$.
(2) *If* $\Theta^{\mathrm{v}\underline{\mathrm{R}}}\mid-\ \vdash\ t\colon(\tau^{\mathrm{v}\underline{\mathrm{R}}}\Rightarrow\underline{\mathsf{R}})\multimap\underline{\mathsf{R}}$ *then there exists a term* $\Theta\vdash M\colon\tau$ *such that* $\Theta^{\mathrm{v}\underline{\mathrm{R}}}\mid-\ \vdash M^{\mathrm{v}\underline{\mathrm{R}}}=t\colon(\tau^{\mathrm{v}\underline{\mathrm{R}}}\Rightarrow\underline{\mathsf{R}})\multimap\underline{\mathsf{R}}$.

Theorem 4.4 (Full completeness of $(\cdot)^{\mathrm{n}\underline{\mathrm{R}}}$)**.** *Suppose* $\underline{\mathsf{R}}$ *is either: (i) a computation-type constant different from* $\underline{\alpha}$*, for every simply-typed* λ*-calculus type constant* α*; or (ii) the type* $\underline{\mathsf{I}}$*. Then:*

(1) *If* $\Theta\vdash M,N\colon\tau$ *and* $\Theta^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}\mid-\ \vdash\ M^{\mathrm{n}\underline{\mathrm{R}}}=N^{\mathrm{n}\underline{\mathrm{R}}}\colon\tau^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}$ *then* $\Theta\ \vdash\ t=_{\beta\eta}u\colon\tau$.
(2) *If* $\Theta^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}\mid-\ \vdash\ t\colon\tau^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}$ *then there exists a term* $\Theta\vdash M\colon\tau$ *such that* $\Theta^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}\mid-\ \vdash M^{\mathrm{n}\underline{\mathrm{R}}}=t\colon\tau^{\mathrm{n}\underline{\mathrm{R}}}\multimap\underline{\mathsf{R}}$.

Theorems 4.3 and 4.4 are analogous to full completeness results obtained by Hasegawa for the linear-use CPS translations into ILL. In [Has02] he proves full completeness for the call-by-value linear-use CPS translation of Moggi's computational λ-calculus [Mog89] into ILL. A similar result holds for the call-by-name translation of [Has04] restricted to the simply-typed λ-calculus (private communication). In both cases, Hasegawa considers translations in which $\underline{\mathsf{R}}$ is taken to be a computation-type constant.

We remark that, in the case that $\underline{\mathsf{R}}$ is a computation-type constant, Theorems 4.3 and 4.4 follow as a consequence of Hasegawa's full completeness results for the translations into ILL. This is because, even though the inclusion of EEC in ILL is neither complete (faithful) nor full, it is sound (preserves equalities) [EMS12]. Hence, for any fully complete translation into ILL that factors through this inclusion, such as the linear-use CPS translations, the factoring translation into EEC is also fully complete. A little thought shows that a similar style of argument cannot be used to derive Hasegawa's results as a consequence of Theorems 4.3 and 4.4. Thus, full completeness with respect to ILL seems a strictly stronger property than full completeness with respect to EEC. Nevertheless, even though Theorems 4.3 and 4.4, in the case that $\underline{\mathsf{R}}$ is a computation-type constant, follow from Hasegawa's results (and not vice-versa), our method of proof is different, and of interest in its own right — see below.

Furthermore, Theorems 4.3 and 4.4 extend Hasegawa's result in a different direction. They apply also when the type $\underline{\mathsf{I}}$ is used for $\underline{\mathsf{R}}$. In the case of the call-by-value translation, this property distinguishes between the translations into EEC and ILL. Indeed, the call-by-value linear-use CPS translation into ILL is not complete if $\underline{\mathsf{I}}$ is used for $\underline{\mathsf{R}}$. A simple counterexample is given by the two λ-calculus terms (3.1) and (3.2), which translate to terms:

$$f\colon (1 \to 1)^{\mathrm{v}\underline{\mathsf{R}}},\ g\colon (1 \to 1)^{\mathrm{v}\underline{\mathsf{R}}} \vdash ((\lambda x\colon 1.\ \lambda y\colon 1.\ *)(f*)(g*))^{\mathrm{v}\underline{\mathsf{R}}}\colon !(1^{\mathrm{v}\underline{\mathsf{R}}})$$
$$f\colon (1 \to 1)^{\mathrm{v}\underline{\mathsf{R}}},\ g\colon (1 \to 1)^{\mathrm{v}\underline{\mathsf{R}}} \vdash ((\lambda x\colon 1.\ \lambda y\colon 1.\ *)(g*)(f*))^{\mathrm{v}\underline{\mathsf{R}}}\colon !(1^{\mathrm{v}\underline{\mathsf{R}}})\ .$$

Noting that $(1 \to 1)^{\mathrm{v}\underline{\mathsf{R}}} = 1 \to ((1 \Rightarrow \underline{\mathsf{I}}) \multimap \underline{\mathsf{I}})$, which is isomorphic, in EEC and (hence) in ILL, to $\underline{\mathsf{I}}$; and $!(1^{\mathrm{v}\underline{\mathsf{R}}}) = !1$, which is also isomorphic to $\underline{\mathsf{I}}$, on can calculuate that the two translated terms are transported along these isomorphisms to:

$$f\colon \underline{\mathsf{I}},\ g\colon \underline{\mathsf{I}} \mid - \vdash \text{let } \top \text{ be } f \text{ in let } \top \text{ be } g \text{ in } \top\colon \underline{\mathsf{I}}$$
$$f\colon \underline{\mathsf{I}},\ g\colon \underline{\mathsf{I}} \mid - \vdash \text{let } \top \text{ be } g \text{ in let } \top \text{ be } f \text{ in } \top\colon \underline{\mathsf{I}}\ .$$

These terms are equal in ILL but not in EEC. (This is reminiscent of the fact that the cbv translation $(\cdot)^{\mathrm{v}}$ of Section 3, when taken into ILL, enforces the commutativity of effects [BW96]; but not identical, because $(\cdot)^{\mathrm{v}\underline{\mathsf{R}}}$ is not, in general, isomorphic to $(\cdot)^{\mathrm{v}}$.)

Our proof of Theorems 4.3 and 4.4 goes via factoring the $(\cdot)^{\mathrm{v}\underline{\mathsf{R}}}$ and $(\cdot)^{\mathrm{n}\underline{\mathsf{R}}}$ through a single generic linear-use CPS translation of the entire enriched effect calculus into itself. This translation, which is the main contribution of the paper, is presented in the next section.

5. Generic linear-use CPS self-translation of EEC

The generic linear-use CPS translation, from EEC to itself, maps a value type A to a value type $\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}$ and a computation type $\underline{\mathsf{A}}$ to a computation type $\underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}$, as defined in Figure 6. Note that the translation of a computation type $\underline{\mathsf{A}}$ as a computation type, $\underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}$, is defined prior to its translation as a value type, $\underline{\mathsf{A}}^{\mathcal{V}_{\underline{\mathsf{R}}}}$. Note also that, in the case that the result type $\underline{\mathsf{R}}$ is a computation-type constant, it is given special treatment. Otherwise it is translated

$$
\begin{aligned}
\alpha^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \alpha & \underline{\alpha}^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \begin{cases} \underline{\alpha} & \text{if } \underline{\alpha} \neq \underline{\mathsf{R}} \\ \underline{\mathsf{I}} & \text{if } \underline{\alpha} = \underline{\mathsf{R}} \end{cases} \\
1^{\mathcal{V}_{\underline{\mathsf{R}}}} &= 1 & \underline{1}^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \underline{0} \\
(\mathsf{A} \times \mathsf{B})^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \times \mathsf{B}^{\mathcal{V}_{\underline{\mathsf{R}}}} & (\underline{\mathsf{A}} \,\&\, \underline{\mathsf{B}})^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \oplus \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \\
(\mathsf{A} \to \mathsf{B})^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \to \mathsf{B}^{\mathcal{V}_{\underline{\mathsf{R}}}} & (\mathsf{A} \Rightarrow \underline{\mathsf{B}})^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \,!(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \\
\underline{\mathsf{A}}^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}} & \underline{\mathsf{I}}^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \underline{\mathsf{R}} \\
(\underline{\mathsf{A}} \multimap \underline{\mathsf{B}})^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} & (\,!\mathsf{A})^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}} \\
& & (\,!\mathsf{A} \otimes \underline{\mathsf{B}})^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \\
& & \underline{0}^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \underline{1} \\
& & (\underline{\mathsf{A}} \oplus \underline{\mathsf{B}})^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \,\&\, \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}
\end{aligned}
$$

Figure 6: Linear-use CPS translation of EEC types.

$$
\begin{aligned}
z^{\mathcal{C}_{\underline{\mathsf{R}}}} &= k_z \\
\underline{*}^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \underline{?_{\mathsf{D}}}(k_z) \\
\underline{\langle t, u \rangle}^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \underline{\text{case}}\, k_z \,\text{of}\, (\underline{\text{inl}}(k_x).\, t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_x/k_z];\, \underline{\text{inr}}(k_y).\, u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_y/k_z]) \\
\underline{\text{fst}}(t)^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[\underline{\text{inl}}(k_z)/k_z] \\
\underline{\text{snd}}(t)^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[\underline{\text{inr}}(k_z)/k_z] \\
(\underline{\lambda} x : \mathsf{A}.\, t)^{\mathcal{C}_{\underline{\mathsf{R}}}} &= \text{let}\, !x \otimes h \,\text{be}\, k_z \,\text{in}\, t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[h/k_z] \\
(s\,\underline{(t)})^{\mathcal{C}_{\underline{\mathsf{R}}}} &= s^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[!(t^{\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes k_z \,/\, k_z] \\
(\text{let}\, \top \,\text{be}\, t \,\text{in}\, u)^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[u^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[k_z] \,/\, k_z] \\
(\text{let}\, !x \,\text{be}\, t \,\text{in}\, u)^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[(\underline{\lambda} x : \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\; u^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[k_z]) \,/\, k_z] \\
(!t \otimes u)^{\mathcal{C}_{\underline{\mathsf{R}}}} &= u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z\,\underline{(t^{\mathcal{V}_{\underline{\mathsf{R}}}})} \,/\, k_z] \\
(\text{let}\, !x \otimes y \,\text{be}\, s \,\text{in}\, t)^{\mathcal{C}_{\underline{\mathsf{R}}}} &= s^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[(\underline{\lambda} x : \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z/k_y]) \,/\, k_z] \\
(\underline{?}(t))^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[\underline{*} \,/\, k_z] \\
(\underline{\text{inl}}(t))^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[\underline{\text{fst}}(k_z) \,/\, k_z] \\
(\underline{\text{inr}}(t))^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[\underline{\text{snd}}(k_z) \,/\, k_z] \\
(\underline{\text{case}}\, s \,\text{of}\, (\underline{\text{inl}}(x).\, t;\, \underline{\text{inr}}(y).\, u))^{\mathcal{C}_{\underline{\mathsf{R}}}} &= s^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[\underline{\langle} t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z/k_x],\, u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z/k_y]\underline{\rangle} \,/\, k_z] \\
(s[t])^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[s^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[k_z] \,/\, k_z]
\end{aligned}
$$

Figure 7: Linear-use CPS translation of computation terms.

$$
\begin{aligned}
x^{\mathcal{V}_{\underline{\mathsf{R}}}} &= x \\
*^{\mathcal{V}_{\underline{\mathsf{R}}}} &= * \\
\langle t, u\rangle^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \langle t^{\mathcal{V}_{\underline{\mathsf{R}}}}, t^{\mathcal{V}_{\underline{\mathsf{R}}}}\rangle \\
(\mathrm{fst}(t))^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \mathrm{fst}(t^{\mathcal{V}_{\underline{\mathsf{R}}}}) \\
(\mathrm{snd}(t))^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \mathrm{snd}(t^{\mathcal{V}_{\underline{\mathsf{R}}}}) \\
(\lambda x\colon \mathsf{A}.\, t)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda x\colon \mathsf{A}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
(t(u))^{\mathcal{V}_{\underline{\mathsf{R}}}} &= t^{\mathcal{V}_{\underline{\mathsf{R}}}}\,(u^{\mathcal{V}_{\underline{\mathsf{R}}}}) \\
\underline{*}^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{0}.\; \underline{?}_{\underline{\mathsf{R}}}(k) \\
\underline{\langle t, u\rangle}^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \oplus \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; \underline{\mathrm{case}}\, k\, \mathrm{of}\, (\underline{\mathrm{inl}}(k_x).\, t^{\mathcal{V}_{\underline{\mathsf{R}}}}[k_x];\, \underline{\mathrm{inr}}(k_y).\, u^{\mathcal{V}_{\underline{\mathsf{R}}}}[k_y]) \\
\underline{\mathrm{fst}}(t)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{\mathrm{inl}}(k)] \\
\underline{\mathrm{snd}}(t)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{\mathrm{inr}}(k)] \\
(\underline{\lambda} x\colon \mathsf{A}.\, t)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon !\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \otimes \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; \mathrm{let}\; !x \otimes h\; \mathrm{be}\; k\; \mathrm{in}\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[h] \\
(s\,\underline{(t)})^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; s^{\mathcal{V}_{\underline{\mathsf{R}}}}[!(t^{\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes k] \\
\top^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{R}}.\; k \\
(\mathrm{let}\; \top\; \mathrm{be}\; t\; \mathrm{in}\; u)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[u^{\mathcal{V}_{\underline{\mathsf{R}}}}[k]] \\
(!t)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}.\; k\,\underline{(t^{\mathcal{V}_{\underline{\mathsf{R}}}})} \\
(\mathrm{let}\; !x\; \mathrm{be}\; t\; \mathrm{in}\; u)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\; u^{\mathcal{V}_{\underline{\mathsf{R}}}}[k]] \\
(!t \otimes u)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; u^{\mathcal{V}_{\underline{\mathsf{R}}}}[k\,\underline{(t^{\mathcal{V}_{\underline{\mathsf{R}}}})}] \\
(\mathrm{let}\; !x \otimes y\; \mathrm{be}\; s\; \mathrm{in}\; t)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{C}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; s^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{C}_{\underline{\mathsf{R}}}}[k/k_y]] \\
(\underline{?}(t))^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{*}] \\
(\underline{\mathrm{inl}}(t))^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \,\&\, \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{\mathrm{fst}}(k)] \\
(\underline{\mathrm{inr}}(t))^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \,\&\, \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{\mathrm{snd}}(k)] \\
(\underline{\mathrm{case}}\, s\, \mathrm{of}\, (\underline{\mathrm{inl}}(x).\, t;\, \underline{\mathrm{inr}}(y).\, u))^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{C}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; s^{\mathcal{V}_{\underline{\mathsf{R}}}}[\underline{\langle} t^{\mathcal{C}_{\underline{\mathsf{R}}}}[k/k_x],\; u^{\mathcal{C}_{\underline{\mathsf{R}}}}[k/k_y]\underline{\rangle}] \\
(\lambda^{\circ} z\colon \underline{\mathsf{A}}.\, t)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{C}_{\underline{\mathsf{R}}}}[k/k_z] \\
(s[t])^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\; t^{\mathcal{V}_{\underline{\mathsf{R}}}}[s^{\mathcal{V}_{\underline{\mathsf{R}}}}[k]]
\end{aligned}
$$

Figure 8: Linear-use CPS translation of value terms.

in the same way as any other type. This means that, when $\underline{\mathsf{R}}$ is either a computation-type constant or $\underline{\mathsf{I}}$, we obtain the complementary equations $\underline{\mathsf{R}}^{\mathcal{C}_{\underline{\mathsf{R}}}} = \underline{\mathsf{I}}$ and $\underline{\mathsf{I}}^{\mathcal{C}_{\underline{\mathsf{R}}}} = \underline{\mathsf{R}}$, exhibiting the computation types $\underline{\mathsf{R}}$ and $\underline{\mathsf{I}}$ as a *dual pair*. Other examples of dual pairs are: $\underline{1}$ and $\underline{0}$; $\underline{\mathsf{A}} \,\&\, \underline{\mathsf{B}}$ and $\underline{\mathsf{A}} \oplus \underline{\mathsf{B}}$; $\mathsf{A} \Rightarrow \underline{\mathsf{B}}$ and $!\mathsf{A} \otimes \underline{\mathsf{B}}$; and $\underline{\alpha}$ (for $\underline{\alpha} \neq \underline{\mathsf{R}}$) with itself. Thus the only computation types without a dual (in this simple sense) are those of the form $!\underline{\mathsf{A}}$. The reason that such dual pairs arise in the translation is that the translation acts *contravariantly* on computation types, in a sense which will be made clear below, but which is already

implicit in the identity $(\underline{\mathsf{A}} \multimap \underline{\mathsf{B}})^{\mathcal{V}_{\mathbb{R}}} = \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}}$. For this reason, each computation type is translated to a computation type that possesses the dual universal property to its own. The contravariance of the computation-type translation also underlies the identity $\underline{\mathsf{A}}^{\mathcal{V}_{\mathbb{R}}} = \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}} \multimap \underline{\mathsf{R}}$, which "negates" the computation-type translation of a computation type in order to bring it into the *covariant* world of value-type translations. We remark that in the conference version of this paper [EMS10], this identity held only up to isomorphism, leading to syntactic complications. The implementation of the equality as a syntactic identity, in Figure 6, is possible in the present paper, because we distinguish between value-type and computation-type products and between value- and computation-type function spaces.

To define the translation of terms, we translate a typing judgement $\Gamma \mid - \vdash t \colon \mathsf{A}$ as:

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash t^{\mathcal{V}_{\mathbb{R}}} \colon \mathsf{A}^{\mathcal{V}_{\mathbb{R}}} \ ,$$

where $\Gamma^{\mathcal{V}_{\mathbb{R}}}$ is the context obtained by applying $(-)^{\mathcal{V}_{\mathbb{R}}}$ to every type in Γ. A typing judgement $\Gamma \mid z \colon \underline{\mathsf{A}} \vdash t \colon \underline{\mathsf{B}}$ is translated to:

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid k_z \colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \vdash t^{\mathcal{C}_{\mathbb{R}}} \colon \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}} \ .$$

The change of direction here is the contravariance we referred to above. The translations are given in Figures 8 and 7 respectively. In these figures, each line corresponds to one of the typing rules in Figure 1, and the type and term names are taken from these rules. Observe that each typing rule that mentions Δ has two cases: one, in Figure 8, for empty stoup in Figure 7, and one for non-empty stoup. Also note that, in Figure 7, we always use $z \colon \underline{\mathsf{D}}$ for the content of a non-empty stoup called Δ in Figure 1. We remark that, because we have the identity $(\underline{\mathsf{A}} \multimap \underline{\mathsf{B}})^{\mathcal{V}_{\mathbb{R}}} = \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}}$, the translations are simpler than those given in the conference version of the paper [EMS10], which involved specified isomorphisms in lieu of the identity.

The remainder of the section is devoted to establishing properties of the self-translation. As a first observation, we note that if $\Gamma \mid - \vdash t \colon \mathsf{A}$, where x is not contained in Γ, then the terms, appearing in each of the translated judgements (cf. Proposition 2.1)

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash t^{\mathcal{V}_{\mathbb{R}}} \colon \mathsf{A}^{\mathcal{V}_{\mathbb{R}}} \qquad \Gamma^{\mathcal{V}_{\mathbb{R}}}, x \colon \mathsf{B}^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash t^{\mathcal{V}_{\mathbb{R}}} \colon \mathsf{A}^{\mathcal{V}_{\mathbb{R}}} \ ,$$

are identical (as the notation suggests). Similarly, if $\Gamma \mid z \colon \underline{\mathsf{A}} \vdash t \colon \underline{\mathsf{B}}$, where again x is not in Γ, then the two terms

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid k_z \colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \vdash t^{\mathcal{C}_{\mathbb{R}}} \colon \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}} \qquad \Gamma^{\mathcal{V}_{\mathbb{R}}}, x \colon \mathsf{C}^{\mathcal{V}_{\mathbb{R}}} \mid k_z \colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \vdash t^{\mathcal{C}_{\mathbb{R}}} \colon \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}}$$

are identical. These observations are easily seen to hold by a straightforward induction on the structure of t.

The interaction between the self-translation and substitution is more subtle. Each of the two cases of Proposition 2.2 splits into two subcases, one for empty Δ, and one for non-empty Δ, resulting in the four cases considered in the proposition below.

Proposition 5.1 (Substitution)**.**

(1) *If* $\Gamma, x \colon \mathsf{A} \mid - \vdash t \colon \mathsf{B}$ *and* $\Gamma \mid - \vdash u \colon \mathsf{A}$ *then* $\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash (t[u/x])^{\mathcal{V}_{\mathbb{R}}} = t^{\mathcal{V}_{\mathbb{R}}}[u^{\mathcal{V}_{\mathbb{R}}} / x] \colon \mathsf{B}^{\mathcal{V}_{\mathbb{R}}}$.

(2) *If* $\Gamma, x \colon \mathsf{A} \mid z \colon \underline{\mathsf{D}} \vdash t \colon \underline{\mathsf{B}}$ *and* $\Gamma \mid - \vdash u \colon \mathsf{A}$ *then*

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid k_z \colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \vdash (t[u/x])^{\mathcal{C}_{\mathbb{R}}} = t^{\mathcal{C}_{\mathbb{R}}}[u^{\mathcal{V}_{\mathbb{R}}} / x] \colon \underline{\mathsf{D}}^{\mathcal{C}_{\mathbb{R}}} \ .$$

(3) *If* $\Gamma \mid x \colon \underline{\mathsf{A}} \vdash t \colon \underline{\mathsf{B}}$ *and* $\Gamma \mid - \vdash u \colon \underline{\mathsf{A}}$ *then*

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash (t[u/x])^{\mathcal{V}_{\mathbb{R}}} = \lambda^{\circ} k \colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}}.\ u^{\mathcal{V}_{\mathbb{R}}}[t^{\mathcal{C}_{\mathbb{R}}}[k / k_x]] \colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \multimap \underline{\mathsf{R}} \ .$$

(4) *If* $\Gamma \mid x\colon \underline{\mathsf{A}} \vdash t\colon \underline{\mathsf{B}}$ *and* $\Gamma \mid z\colon \underline{\mathsf{D}} \vdash u\colon \underline{\mathsf{A}}$ *then*

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid k_z\colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \vdash (t[u/x])^{\mathcal{C}_{\mathbb{R}}} = u^{\mathcal{C}_{\mathbb{R}}}[(t^{\mathcal{C}_{\mathbb{R}}}[k_z\,/k_x])\,/\,k_z]\colon \underline{\mathsf{D}}^{\mathcal{C}_{\mathbb{R}}}\ .$$

Proof. By induction on t.

Statements 1 and 2 are proved simultaneously. For example, if t is $\lambda^{\circ} z\colon \underline{\mathsf{B}}_1.\ t'$, where $\underline{\mathsf{B}}$ is $\underline{\mathsf{B}}_1 \multimap \underline{\mathsf{B}}_2$, then statement 1 applies, and we must show that $((\lambda^{\circ} z\colon \underline{\mathsf{B}}_1.\ t')[u/x])^{\mathcal{V}_{\mathbb{R}}} = (\lambda^{\circ} z\colon \underline{\mathsf{B}}_1.\ t')^{\mathcal{V}_{\mathbb{R}}}[u^{\mathcal{V}_{\mathbb{R}}}\,/\,x]$. The induction hypothesis, given by statement 2, is $(t'[u/x])^{\mathcal{C}_{\mathbb{R}}} = (t')^{\mathcal{C}_{\mathbb{R}}}[u^{\mathcal{V}_{\mathbb{R}}}\,/\,x]$. And indeed:

$$\begin{aligned}
((\lambda^{\circ} z\colon \underline{\mathsf{B}}_1.\ t')[u/x])^{\mathcal{V}_{\mathbb{R}}} &= (\lambda^{\circ} z\colon \underline{\mathsf{B}}_1.\ t'[u/x])^{\mathcal{V}_{\mathbb{R}}} \\
&= \lambda^{\circ} k_z\colon \underline{\mathsf{B}}_2^{\mathcal{C}_{\mathbb{R}}}.\ (t'[u/x])^{\mathcal{C}_{\mathbb{R}}} \\
&= \lambda^{\circ} k_z\colon \underline{\mathsf{B}}_2^{\mathcal{C}_{\mathbb{R}}}.\ (t')^{\mathcal{C}_{\mathbb{R}}}[u^{\mathcal{V}_{\mathbb{R}}}\,/\,x] \qquad \text{by induction hypothesis} \\
&= (\lambda^{\circ} z\colon \underline{\mathsf{B}}_1.\ t')^{\mathcal{V}_{\mathbb{R}}}[u^{\mathcal{V}_{\mathbb{R}}}\,/\,x]\ .
\end{aligned}$$

We illustrate the proof of statement 3 in the case that t is $\underline{\mathrm{case}}\, t'$ of $(\underline{\mathrm{inl}}(y).\, t_1;\ \underline{\mathrm{inr}}(z).\, t_2)$, where $\Gamma \mid x\colon \underline{\mathsf{A}} \vdash t'\colon \underline{\mathsf{C}}_1 \oplus \underline{\mathsf{C}}_2$, and $\Gamma \mid y\colon \underline{\mathsf{C}}_1 \vdash t_1\colon \underline{\mathsf{B}}$, and $\Gamma \mid z\colon \underline{\mathsf{C}}_2 \vdash t_2\colon \underline{\mathsf{B}}$. Then:

$$\begin{aligned}
(t[u/x])^{\mathcal{V}_{\mathbb{R}}} &= (\underline{\mathrm{case}}\, t'[u/x] \text{ of } (\underline{\mathrm{inl}}(y).\, t_1;\ \underline{\mathrm{inr}}(z).\, t_2))^{\mathcal{V}_{\mathbb{R}}} \\
&= \lambda^{\circ} k\colon \underline{\mathsf{B}}.\ (t'[u/x])^{\mathcal{V}_{\mathbb{R}}}[\underline{\langle} t_1^{\mathcal{C}_{\mathbb{R}}}[k/k_y], t_2^{\mathcal{C}_{\mathbb{R}}}[k/k_z]\underline{\rangle}] \\
&= \lambda^{\circ} k\colon \underline{\mathsf{B}}.\ u^{\mathcal{V}_{\mathbb{R}}}[(t')^{\mathcal{C}_{\mathbb{R}}}[\underline{\langle} t_1^{\mathcal{C}_{\mathbb{R}}}[k/k_y], t_2^{\mathcal{C}_{\mathbb{R}}}[k/k_z]\underline{\rangle}\,/\,k_x]] \qquad \text{by induction hypothesis} \\
&= \lambda^{\circ} k\colon \underline{\mathsf{B}}.\ u^{\mathcal{V}_{\mathbb{R}}}[(\underline{\mathrm{case}}\, t' \text{ of } (\underline{\mathrm{inl}}(y).\, t_1;\ \underline{\mathrm{inr}}(z).\, t_2))^{\mathcal{C}_{\mathbb{R}}}[k/k_x]] \\
&= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}}.\ u^{\mathcal{V}_{\mathbb{R}}}[t^{\mathcal{C}_{\mathbb{R}}}[k/k_x]]\ .
\end{aligned}$$

We omit the proof of statement 4, which is straightforward. □

We now have the machinery necessary to establish the first of the main properties of the self-translation, its equational soundness.

Theorem 5.2 (Soundness)**.**

(1) *If* $\Gamma \mid - \vdash t = u\colon \mathsf{A}$ *then* $\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash t^{\mathcal{V}_{\mathbb{R}}} = u^{\mathcal{V}_{\mathbb{R}}}\colon \mathsf{A}^{\mathcal{V}_{\mathbb{R}}}$.

(2) *If* $\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t = u\colon \underline{\mathsf{B}}$ *then* $\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid k_z\colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \vdash t^{\mathcal{C}_{\mathbb{R}}} = u^{\mathcal{C}_{\mathbb{R}}}\colon \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}}$.

Proof. Define $\Gamma \mid - \vdash t \sim u\colon \mathsf{A}$ to hold if $\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash t^{\mathcal{V}_{\mathbb{R}}} = u^{\mathcal{V}_{\mathbb{R}}}\colon \mathsf{A}^{\mathcal{V}_{\mathbb{R}}}$, and similarly $\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t \sim u\colon \underline{\mathsf{B}}$ to hold if $\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid k_z\colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \vdash t^{\mathcal{C}_{\mathbb{R}}} = u^{\mathcal{C}_{\mathbb{R}}}\colon \underline{\mathsf{A}}^{\mathcal{C}_{\mathbb{R}}}$. Trivially, $\sim$ is a type-respecting equivalence relation. By the compositional definition of $(\cdot)^{\mathcal{V}_{\mathbb{R}}}$ and $(\cdot)^{\mathcal{C}_{\mathbb{R}}}$ it is an α-equivalence respecting congruence. It remains to verify that $\sim$ satisfies the equalities of Figure 2. Once again, every equality in which Δ appears, splits into two cases, one for empty Δ, and one for non-empty Δ. This means that the 24 equalities of Figure 2, give rise to 39 equalities that need verifying. We consider two cases, by way of illustration.

For the first case, suppose $\Gamma \mid - \vdash t\colon \mathsf{A}$ and $\Gamma,\ x\colon \mathsf{A} \mid - \vdash u\colon \underline{\mathsf{B}}$. We show that

$$\Gamma^{\mathcal{V}_{\mathbb{R}}} \mid - \vdash (\text{let } !x \text{ be } !t \text{ in } u)^{\mathcal{V}_{\mathbb{R}}} = (u[t/x])^{\mathcal{V}_{\mathbb{R}}}\colon \underline{\mathsf{B}}^{\mathcal{C}_{\mathbb{R}}} \multimap \underline{\mathsf{R}}\ .$$

For this,

$$\begin{aligned}
(\text{let } !x \text{ be } !t \text{ in } u)^{\mathcal{V}_{\underline{\mathsf{R}}}} &= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\ (!t)^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ u^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[k]] && \\
&= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\ (\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ u^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[k])\,\underline{(t^{\mathcal{V}_{\underline{\mathsf{R}}}})} && \text{def. of } (!t)^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
&= \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\ u^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[t^{\mathcal{V}_{\underline{\mathsf{R}}}}/x]\,[k] && \beta \text{ equality} \\
&= u^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[t^{\mathcal{V}_{\underline{\mathsf{R}}}}/x] && \eta \text{ equality} \\
&= (u[t/x])^{\mathcal{V}_{\underline{\mathsf{R}}}} && \text{Prop. 5.1.1 .}
\end{aligned}$$

For the second case, suppose $\Gamma \,|\, z\colon \underline{\mathsf{D}} \vdash t\colon !\mathsf{A}$ and $\Gamma \,|\, y\colon !\mathsf{A} \vdash u\colon \underline{\mathsf{B}}$. We show that

$$\Gamma \,|\, k_z\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \vdash (\text{let } !x \text{ be } t \text{ in } u[!x/y])^{\mathcal{C}_{\underline{\mathsf{R}}}} = (u[t/y])^{\mathcal{C}_{\underline{\mathsf{R}}}}\colon \underline{\mathsf{D}}^{\mathcal{C}_{\underline{\mathsf{R}}}}\ .$$

For this,

$$\begin{aligned}
(\text{let } !x \text{ be } t \text{ in } u[!x/y])^{\mathcal{C}_{\underline{\mathsf{R}}}} &= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[(\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ (u[!x/y])^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[k_z])\,/\,k_z] && \\
&= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[(\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ (!x)^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z/k_y]])\,/\,k_z] && \text{Prop. 5.1.3} \\
&= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[(\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ (u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z/k_y])\underline{(x^{\mathcal{V}_{\underline{\mathsf{R}}}})})\,/\,k_z] && \text{def. of } (!x)^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
&= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[(\underline{\lambda} x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ (u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z/k_y])\underline{(x)})\,/\,k_z] && \text{def. of } x^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
&= t^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[(u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k_z/k_y])\,/\,k_z] && \eta \text{ equality} \\
&= (u[t/y])^{\mathcal{C}_{\underline{\mathsf{R}}}} && \text{Prop. 5.1.4 .}
\end{aligned}$$

We comment that the second step above, employs the equality

$$(u[!x/y])^{\mathcal{V}_{\underline{\mathsf{R}}}} = \lambda^{\circ} k\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}}.\ (!x)^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[u^{\mathcal{C}_{\underline{\mathsf{R}}}}\,[k\,/\,k_y]]\ ,$$

whose strict derivation from Proposition 5.1.3 invokes the coincidence of the two terms:

$$\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \,|\, k_y\colon \underline{\mathsf{B}} \vdash u\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}} \qquad \Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}}, x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \,|\, k_y\colon \underline{\mathsf{B}} \vdash u\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}\ .$$

Having made this point once, we shall not comment further on such small issues arising from weakening. □

We now come to the central result of the paper: if $\underline{\mathsf{R}}$ is either a computation-type constant or $\underline{\mathsf{I}}$ then the self-translation is involutive up to isomorphism (Theorem 5.4). That is, the translation of the translation of a term is equal, modulo type isomorphism, to the original term. To state the involution property, we first define the required isomorphisms. For each value type A, we define a closed EEC term, $i_{\mathsf{A}}\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}} \to \mathsf{A}$, for which there exists a corresponding closed term $i_{\mathsf{A}}^{-1}\colon \mathsf{A} \to \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}$ such that the equations $\lambda x\colon \mathsf{A}.\ i_{\mathsf{A}}(i_{\mathsf{A}}^{-1}(x)) = \lambda x\colon \mathsf{A}.\ x$ and $\lambda x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}.\ i_{\mathsf{A}^{-1}}(i_{\mathsf{A}}(x)) = \lambda x\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}.\ x$ hold in the EEC equational theory. Similarly, for each computation type $\underline{\mathsf{A}}$, we define a closed EEC term $j_{\underline{\mathsf{A}}}\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{A}}$ that is a linear isomorphism. That is, the inverse is given by a closed term $j_{\underline{\mathsf{A}}}^{-1}\colon \underline{\mathsf{A}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}$ such that the equations asserting the mutual inverse properties again hold. The families of terms i_{A} and $j_{\underline{\mathsf{A}}}$ are mutually defined by induction on their types in Figure 9. Note that, for a computation type $\underline{\mathsf{A}}$, the linear isomorphism $j_{\underline{\mathsf{A}}}$ is defined first, and the definition of $i_{\underline{\mathsf{A}}}$ depends on it. Note also that the clauses for function types require the inverses of previously defined terms, which, since they are inverses, are uniquely determined up to provable equality. Their existence is assured by the lemma below, which therefore establishes that Figure 9 is a good definition.

$$
\begin{aligned}
i_\alpha &= \lambda x{:}\alpha.\ x \\
i_1 &= \lambda x{:}1.\ * \\
i_{\mathsf{A}\times\mathsf{B}} &= \lambda z{:}\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}} \times \mathsf{B}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}.\ \langle i_{\mathsf{A}}(\mathrm{fst}(z)), i_{\mathsf{B}}(\mathrm{snd}(z))\rangle \\
i_{\mathsf{A}\to\mathsf{B}} &= \lambda f{:}\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}} \to \mathsf{B}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}.\ \lambda x{:}\mathsf{A}.\ i_{\mathsf{B}}(f(i_{\mathsf{A}}^{-1}(x))) \\
i_{\underline{\mathsf{A}}} &= \lambda h{:}\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ j_{\underline{\mathsf{A}}}\,[h\,[\top]] \\
i_{\underline{\mathsf{A}}\multimap\underline{\mathsf{B}}} &= \lambda h{:}\underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \lambda^\circ x{:}\underline{\mathsf{A}}.\ j_{\underline{\mathsf{B}}}\,[h\,[j_{\underline{\mathsf{A}}}^{-1}\,[x]]] \\
\\
j_{\underline{\alpha}} &= \lambda^\circ z{:}\underline{\alpha}.\ z \\
j_{\underline{1}} &= \lambda^\circ z{:}\underline{1}.\ \underline{*} \\
j_{\underline{\mathsf{A}}\,\&\,\underline{\mathsf{B}}} &= \lambda^\circ z{:}\underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}\,\&\,\underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \underline{\langle}\, j_{\underline{\mathsf{A}}}\,[\underline{\mathrm{fst}}(z)],\ j_{\underline{\mathsf{B}}}\,[\underline{\mathrm{snd}}(z)]\,\underline{\rangle} \\
j_{\mathsf{A}\Rightarrow\underline{\mathsf{B}}} &= \lambda^\circ f{:}\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \underline{\lambda} x{:}\mathsf{A}.\ j_{\underline{\mathsf{B}}}\,[f(\underline{i_{\mathsf{A}}^{-1}(x)})] \\
j_{\underline{\mathsf{I}}} &= \lambda^\circ z{:}\underline{\mathsf{I}}.\ z \\
j_{!\mathsf{A}} &= \lambda^\circ z{:}!(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}.\ \text{let } !x{\otimes}y \text{ be } z \text{ in let } \top \text{ be } y \text{ in } !(i_{\mathsf{A}}(x)) \\
j_{!\mathsf{A}\otimes\underline{\mathsf{B}}} &= \lambda^\circ z{:}!(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \text{let } !x{\otimes}y \text{ be } z \text{ in } !(i_{\mathsf{A}}(x)){\otimes}(j_{\underline{\mathsf{B}}}\,[y]) \\
j_{\underline{0}} &= \lambda^\circ z{:}\underline{0}.\ z \\
j_{\underline{\mathsf{A}}\oplus\underline{\mathsf{B}}} &= \lambda^\circ z{:}\underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} \oplus \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \underline{\mathrm{case}}\, z \,\mathrm{of}\, (\underline{\mathrm{inl}}(x).\ \underline{\mathrm{inl}}(j_{\underline{\mathsf{A}}}\,[x]);\ \underline{\mathrm{inr}}(y).\ \underline{\mathrm{inr}}(j_{\underline{\mathsf{B}}}\,[y]))
\end{aligned}
$$

Figure 9: Type isomorphisms for the involution property

Lemma 5.3. *Suppose* $\underline{\mathsf{R}}$ *is either a computation-type constant or* $\underline{\mathsf{I}}$*. Then each term* $i_{\mathsf{A}}\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}} \to \mathsf{A}$ *is an isomorphism, and each* $j_{\underline{\mathsf{A}}}\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{A}}$ *is a linear isomorphism.*

Proof. The two statements are proved simultaneously by induction on the type, with, in the case of a computation type $\underline{\mathsf{A}}$, the inverse for $j_{\underline{\mathsf{A}}}$ being established before that of $i_{\underline{\mathsf{A}}}$. The assumption that $\underline{\mathsf{R}}$ is either a computation-type constant or $\underline{\mathsf{I}}$ implies that $\underline{\mathsf{R}}^{\mathcal{C}_{\underline{\mathsf{R}}}} = \underline{\mathsf{I}}$, and this fact is used frequently in the proof. We consider just two illustrative cases: $i_{\underline{\mathsf{A}}}$ and $j_{!\mathsf{A}}$.

In the case of $i_{\underline{\mathsf{A}}}$, we have $\underline{\mathsf{A}}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}} = (\underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}})^{\mathcal{V}_{\underline{\mathsf{R}}}} = \underline{\mathsf{R}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} = \underline{\mathsf{I}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}$, and the inverse $i_{\underline{\mathsf{A}}}^{-1}\colon \underline{\mathsf{A}} \to (\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}})$ is defined by

$$i_{\underline{\mathsf{A}}}^{-1} \;=\; \lambda x{:}\underline{\mathsf{A}}.\ \lambda^\circ z{:}\underline{\mathsf{I}}.\ \text{let } \top \text{ be } z \text{ in } j_{\underline{\mathsf{A}}}^{-1}[x]\ .$$

Then we have (using the obvious definition for composition):

$$
\begin{aligned}
i_{\underline{\mathsf{A}}}^{-1} \circ i_{\underline{\mathsf{A}}} &= \lambda h{:}\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \lambda^\circ z{:}\underline{\mathsf{I}}.\ \text{let } \top \text{ be } z \text{ in } j_{\underline{\mathsf{A}}}^{-1}[j_{\underline{\mathsf{A}}}\,[h\,[\top]]] \\
&= \lambda h{:}\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \lambda^\circ z{:}\underline{\mathsf{I}}.\ \text{let } \top \text{ be } z \text{ in } h\,[\top] \qquad \text{by induction hypothesis} \\
&= \lambda h{:}\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ \lambda^\circ z{:}\underline{\mathsf{I}}.\ h\,[z] \\
&= \lambda h{:}\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}.\ h\ ,
\end{aligned}
$$

and the verification that $i_{\underline{\mathsf{A}}} \circ i_{\underline{\mathsf{A}}}^{-1} = \lambda x{:}\underline{\mathsf{A}}.\ x$ is similarly straightforward.

In the case of $j_{!\mathsf{A}}$, we have $(!\mathsf{A})^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} = !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}$ and the inverse $j_{!\mathsf{A}}^{-1}\colon !\mathsf{A} \multimap !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}$ is defined by

$$j_{!\mathsf{A}}^{-1} \;=\; \lambda^\circ w{:}!\mathsf{A}.\ \text{let } !x \text{ be } w \text{ in } !(i_{\mathsf{A}}^{-1}(x)){\otimes}\top\ . \tag{5.1}$$

Then:

$$\begin{aligned}
& j_{!\mathsf{A}}^{-1} \circ j_{!\mathsf{A}} \\
&\quad = \lambda^{\circ} z\colon !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}.\ \text{let } !x' \text{ be } (\text{let } !x \otimes y \text{ be } z \text{ in let } \top \text{ be } y \text{ in } !(i_{\mathsf{A}}(x))) \text{ in } !(i_{\mathsf{A}}^{-1}(x')) \otimes \top \\
&\quad = \lambda^{\circ} z\colon !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}.\ \text{let } !x \otimes y \text{ be } z \text{ in let } \top \text{ be } y \text{ in let } !x' \text{ be } !(i_{\mathsf{A}}(x)) \text{ in } !(i_{\mathsf{A}}^{-1}(x')) \otimes \top \\
&\quad = \lambda^{\circ} z\colon !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}.\ \text{let } !x \otimes y \text{ be } z \text{ in let } \top \text{ be } y \text{ in } !(i_{\mathsf{A}}^{-1}(i_{\mathsf{A}}(x))) \otimes \top \\
&\quad = \lambda^{\circ} z\colon !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}.\ \text{let } !x \otimes y \text{ be } z \text{ in let } \top \text{ be } y \text{ in } !x \otimes \top \\
&\quad = \lambda^{\circ} z\colon !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}.\ \text{let } !x \otimes y \text{ be } z \text{ in } !x \otimes y \\
&\quad = \lambda^{\circ} z\colon !(\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes \underline{\mathsf{I}}.\ z\ ,
\end{aligned}$$

where the third equality applies the induction hypothesis, and all others, including the rearrangement of "let" expressions in the second equation, justified by the equalities of Figure 2. The verification that $j_{!\mathsf{A}} \circ j_{!\mathsf{A}}^{-1} = \lambda x\colon !\mathsf{A}.\ x$ is straightforward. □

We remark that the main reason for including $\underline{\mathsf{I}}$ as a primitive EEC construct, in the present paper, was to permit the uniform definition of the type isomorphisms, given in Figure 9, which covers both cases of interest: when $\underline{\mathsf{R}}$ is a computation-type constant, and when it is $\underline{\mathsf{I}}$. The alternative would have beeen to have omitted $\underline{\mathsf{I}}$ from the primitive syntax, defining it as !1. Had this been done, we would have obtained: $\underline{\mathsf{R}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} = 1 \Rightarrow \underline{\mathsf{R}}$, in the case that $\underline{\mathsf{R}}$ is a computation-type constant; and $\underline{\mathsf{R}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}} = !1 \otimes (1 \Rightarrow \underline{\mathsf{R}})$, in the case that $\underline{\mathsf{R}}$ is $\underline{\mathsf{I}}$ (i.e., $\underline{\mathsf{R}} = !1$). In both cases, linear isomorphisms between $\underline{\mathsf{R}}$ and $\underline{\mathsf{R}}^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}$ still exist, they can no longer be given uniformly.

In order to state the fundamental involution property enjoyed by the self-translation on EEC, for a context

$$\Gamma\ =\ x_1\colon \mathsf{C}_1, \ldots, x_n\colon \mathsf{C}_n\ ,$$

we introduce the notation $[i^{-1}(\Gamma)]$ for the substitution

$$[\, i_{\mathsf{C}_1}^{-1}(x_1), \ldots, i_{\mathsf{C}_n}^{-1}(x_n) \,/\, x_1, \ldots, x_n \,]\ .$$

Theorem 5.4 (Involution property)**.** *Suppose* $\underline{\mathsf{R}}$ *is either a computation-type constant or* $\underline{\mathsf{I}}$*.*
(1) *If* $\Gamma \mid - \vdash t\colon \mathsf{A}$ *then* $\Gamma \mid - \vdash t = i_{\mathsf{A}}(t^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}})\,[i^{-1}(\Gamma)]\colon \mathsf{A}$.
(2) *If* $\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t\colon \underline{\mathsf{B}}$ *then* $\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t = j_{\underline{\mathsf{B}}}\,[\,t^{\mathcal{C}_{\underline{\mathsf{R}}}\mathcal{C}_{\underline{\mathsf{R}}}}]\,[j_{\underline{\mathsf{A}}}^{-1}[z] \,/\, k_{k_z}]\,[i^{-1}(\Gamma)] : \underline{\mathsf{B}}$.

Proof. The statements are proved simultaneously, by induction on t. There are 41 cases in the proof, one for each of the equations in Figures 7 and 8. By way of illustration, we verify two of them, the second being among the most complex cases in the proof.

For the first case, suppose $\Gamma \mid - \vdash t\colon \mathsf{A}$. We verify that:

$$\Gamma \mid - \vdash\ !t = i_{!\mathsf{A}}((!t)^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}})\,[i^{-1}(\Gamma)]\colon !\mathsf{A}\ .$$

The basic strategy is to first expand the inner $(\cdot)^{\mathcal{V}_{\underline{\mathsf{R}}}}$, then the outer $(\cdot)^{\mathcal{V}_{\underline{\mathsf{R}}}}$, applying the definitions of $i_{!\mathsf{A}}$ and $j_{!\mathsf{A}}$ until the induction hypothesis can be invoked. Between these steps, we use the equalities of Figure 2 to simplify the terms as far as possible. Henceforth, we treat applications of equalities from Figure 2 as trivial. So, in the detailed derivation below, we do not annotate such steps. Nor do we explain obvious expansions of $(\cdot)^{\mathcal{V}_{\underline{\mathsf{R}}}}$ and $(\cdot)^{\mathcal{C}_{\underline{\mathsf{R}}}}$.

$$\begin{aligned}
& i_{!\mathsf{A}}((!t)^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}})\,[i^{-1}(\Gamma)] \\
&\quad = i_{!\mathsf{A}}((\lambda^{\circ} k\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}.\ k\,(\underline{t^{\mathcal{V}_{\underline{\mathsf{R}}}}})^{\mathcal{V}_{\underline{\mathsf{R}}}})\,[i^{-1}(\Gamma)]
\end{aligned}$$

$$
\begin{aligned}
&= i_{!\mathsf{A}}(\lambda^\circ k' : \underline{\mathsf{R}}^{\mathcal{C}_{\mathbb{R}}}.\ (k\ \underline{(t^{\mathcal{V}_{\mathbb{R}}})})^{\mathcal{C}_{\mathbb{R}}}\,[k'/k_k])\,[i^{-1}(\Gamma)] \\
&= i_{!\mathsf{A}}(\lambda^\circ k' : \underline{\mathsf{R}}^{\mathcal{C}_{\mathbb{R}}}.\ k^{\mathcal{C}_{\mathbb{R}}}\,[!(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}})\otimes k_k\,/k_k]\ [k'/k_k])\,[i^{-1}(\Gamma)] \\
&= i_{!\mathsf{A}}(\lambda^\circ k' : \underline{\mathsf{R}}^{\mathcal{C}_{\mathbb{R}}}.\ k_k\,[!(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}})\otimes k_k\,/k_k]\ [k'/k_k])\,[i^{-1}(\Gamma)] \\
&= i_{!\mathsf{A}}(\lambda^\circ k' : \underline{\mathsf{R}}^{\mathcal{C}_{\mathbb{R}}}.\ !(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}})\otimes k')\,[i^{-1}(\Gamma)] \\
&= j_{!\mathsf{A}}[(\lambda^\circ k' : \underline{\mathsf{R}}^{\mathcal{C}_{\mathbb{R}}}.\ !(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}})\otimes k')\,[\top]]\,[i^{-1}(\Gamma)] && \text{def. of } i_{!\mathsf{A}} \\
&= j_{!\mathsf{A}}[!(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}})\otimes\top]\,[i^{-1}(\Gamma)] \\
&= \text{let } !x\otimes y \text{ be } !(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}})\otimes\top \text{ in let } \top \text{ be } y \text{ in } !(i_{\mathsf{A}}(x))\,[i^{-1}(\Gamma)] && \text{def. of } j_{!\mathsf{A}} \\
&= \text{let } \top \text{ be } \top \text{ in } !(i_{\mathsf{A}}(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}))\,[i^{-1}(\Gamma)] \\
&= !(i_{\mathsf{A}}(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}))\,[i^{-1}(\Gamma)] \\
&= !(i_{\mathsf{A}}(t^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}})\,[i^{-1}(\Gamma)]) \\
&= !t && \text{induction hypothesis .}
\end{aligned}
$$

For the second case, suppose $\Gamma \mid z : \underline{\mathsf{D}} \vdash t : !\mathsf{A}$ and $\Gamma, x : \mathsf{A} \mid - \vdash u : \underline{\mathsf{B}}$. We verify that

$$\Gamma \mid z : \underline{\mathsf{D}} \vdash \text{let } !x \text{ be } t \text{ in } u \;=\; j_{\underline{\mathsf{B}}}\,[(\text{let } !x \text{ be } t \text{ in } u)^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}}]\,[j_{\underline{\mathsf{B}}}^{-1}[z]\,/\,k_{k_z}]\,[i^{-1}(\Gamma)] : \underline{\mathsf{B}}\ .$$

Adopting a similar strategy to above, we obtain:

$$
\begin{aligned}
&j_{\underline{\mathsf{B}}}\,[(\text{let } !x \text{ be } t \text{ in } u)^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}}]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= j_{\underline{\mathsf{B}}}\,[(t^{\mathcal{C}_{\mathbb{R}}}\,[(\underline{\lambda} x : \mathsf{A}^{\mathcal{V}_{\mathbb{R}}}.\ u^{\mathcal{V}_{\mathbb{R}}}\,[k_z])\,/\,k_z])^{\mathcal{C}_{\mathbb{R}}}]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= j_{\underline{\mathsf{B}}}\,[(\underline{\lambda} x : \mathsf{A}^{\mathcal{V}_{\mathbb{R}}}.\ u^{\mathcal{V}_{\mathbb{R}}}\,[k_z])^{\mathcal{C}_{\mathbb{R}}}\,[t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}}\,/\,k_{k_z}]]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] && (5.2) \\
&= j_{\underline{\mathsf{B}}}\,[(\text{let } !x\otimes h \text{ be } k_{k_z} \text{ in } (u^{\mathcal{V}_{\mathbb{R}}}\,[k_z])^{\mathcal{C}_{\mathbb{R}}}\,[h/k_{k_z}])\,[t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}}\,/\,k_{k_z}]]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= j_{\underline{\mathsf{B}}}\,[(\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in } (u^{\mathcal{V}_{\mathbb{R}}}\,[k_z])^{\mathcal{C}_{\mathbb{R}}}\,[h/k_{k_z}])]\,[j_{\underline{\mathsf{B}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= j_{\underline{\mathsf{B}}}\,[(\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in } (k_z^{\mathcal{C}_{\mathbb{R}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}\,[k_{k_z}]\,/\,k_{k_z}])\,[h/k_{k_z}])]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= j_{\underline{\mathsf{B}}}\,[(\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in } (k_{k_z}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}\,[k_{k_z}]\,/\,k_{k_z}])\,[h/k_{k_z}])]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= j_{\underline{\mathsf{B}}}\,[(\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in } u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}\,[k_{k_z}]\,[h/k_{k_z}])]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= j_{\underline{\mathsf{B}}}\,[(\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in } u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}\,[h])]\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= (\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in } j_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}\,[h]])\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= (\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in let } \top \text{ be } h \text{ in } j_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}\,[\top]])\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] \\
&= (\text{let } !x\otimes h \text{ be } t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}} \text{ in let } \top \text{ be } h \text{ in } i_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}])\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)] && (5.3) \\
&= \text{let } !x\otimes h \text{ be } (t^{\mathcal{C}_{\mathbb{R}}\mathcal{C}_{\mathbb{R}}}\,[j_{\underline{\mathsf{D}}}^{-1}[z]/k_{k_z}]\,[i^{-1}(\Gamma)]) \text{ in let } \top \text{ be } h \text{ in } (i_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}]\,[i^{-1}(\Gamma)]) \\
&= \text{let } !x\otimes h \text{ be } j_{!\mathsf{A}}^{-1}[t] \text{ in let } \top \text{ be } h \text{ in } (i_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}]\,[i^{-1}(\Gamma)]) && (5.4) \\
&= \text{let } !x\otimes h \text{ be } (\text{let } !x \text{ be } t \text{ in } !(i_{\mathsf{A}}^{-1}(x))\otimes\top) \text{ in let } \top \text{ be } h \text{ in } (i_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}]\,[i^{-1}(\Gamma)]) && (5.5) \\
&= \text{let } !x \text{ be } t \text{ in let } !x\otimes h \text{ be } !(i_{\mathsf{A}}^{-1}(x))\otimes\top \text{ in let } \top \text{ be } h \text{ in } (i_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}]\,[i^{-1}(\Gamma)]) \\
&= \text{let } !x \text{ be } t \text{ in let } \top \text{ be } \top \text{ in } (i_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}]\,[i^{-1}(\Gamma)]\,[i_{\mathsf{A}}^{-1}(x)\,/\,x]) \\
&= \text{let } !x \text{ be } t \text{ in } (i_{\underline{\mathsf{B}}}\,[u^{\mathcal{V}_{\mathbb{R}}\mathcal{V}_{\mathbb{R}}}]\,[i^{-1}(\Gamma, x : \mathsf{A})])
\end{aligned}
$$

$$= \text{let } !x \text{ be } t \text{ in } u \ . \tag{5.6}$$

Here, (5.2) is by Proposition 5.1.4, (5.3) is by definition of $j_{\underline{\mathsf{B}}}$, (5.4) applies the induction hypothesis for t, (5.5) expands $j^{-1}_{!\mathsf{A}}$ using (5.1), and (5.6) applies the induction hypothesis for t (which is applicable only at this point in the argument, because t is typed relative to the context $\Gamma, x\colon \mathsf{A}$ rather than Γ). □

We end the present section by applying Theorem 5.4 to derive the full completeness of the self-translation.

Theorem 5.5 (Full completeness of self-translation)**.** *Suppose* $\underline{\mathsf{R}}$ *is either a computation-type constant or* $\underline{\mathsf{I}}$*.*

(1) *If* $\Gamma \mid\!- \vdash t, u\colon \mathsf{A}$ *and* $\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \mid\!- \vdash t^{\mathcal{V}_{\underline{\mathsf{R}}}} = u^{\mathcal{V}_{\underline{\mathsf{R}}}}\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}$ *then* $\Gamma \mid\!- \vdash t = u\colon \mathsf{A}$*.*
(2) *If* $\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \mid\!- \vdash t\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}$ *then there exists* $\Gamma \mid\!- \vdash u\colon \mathsf{A}$ *such that* $\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \mid\!- \vdash t = u^{\mathcal{V}_{\underline{\mathsf{R}}}}\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}$*.*
(3) *If* $\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t, u\colon \underline{\mathsf{B}}$ *and* $\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \mid k_z\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \vdash t^{\mathcal{C}_{\underline{\mathsf{R}}}} = u^{\mathcal{C}_{\underline{\mathsf{R}}}}\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}$ *then* $\Gamma \mid z\colon \underline{\mathsf{A}} \vdash t = u\colon \underline{\mathsf{B}}$*.*
(4) *If* $\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \mid k_z\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \vdash t\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}$ *then there exists* $\Gamma \mid z\colon \underline{\mathsf{A}} \vdash u\colon \underline{\mathsf{B}}$ *such that* $\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \mid k_z\colon \underline{\mathsf{B}}^{\mathcal{C}_{\underline{\mathsf{R}}}} \vdash t = u^{\mathcal{C}_{\underline{\mathsf{R}}}}\colon \underline{\mathsf{A}}^{\mathcal{C}_{\underline{\mathsf{R}}}}$*.*

Proof. For statement 1, suppose $\Gamma \mid\!- \vdash t, u\colon \mathsf{A}$ and $t^{\mathcal{V}_{\underline{\mathsf{R}}}} = u^{\mathcal{V}_{\underline{\mathsf{R}}}}$. Then:

$$\begin{aligned} t &= i_{\mathsf{A}}\,(t^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}[i^{-1}(\Gamma)]) && \text{(Theorem 5.4.1)} \\ &= i_{\mathsf{A}}\,(u^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}[i^{-1}(\Gamma)]) && \text{(Theorem 5.2)} \\ &= u && \text{(Theorem 5.4.1)} \ . \end{aligned}$$

For statement 2, suppose $\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}} \mid\!- \vdash t\colon \mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}$. Define $u = i_{\mathsf{A}}(t^{\mathcal{V}_{\underline{\mathsf{R}}}}[i^{-1}(\Gamma)])$. Then:

$$\begin{aligned} u^{\mathcal{V}_{\underline{\mathsf{R}}}} &= i_{\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}}(u^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}[i^{-1}(\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}})]) && \text{(Theorem 5.4.1)} \\ &= i_{\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}}((i^{-1}_{\underline{\mathsf{A}}}(u\,[i(\Gamma)]))^{\mathcal{V}_{\underline{\mathsf{R}}}}[i^{-1}(\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}})]) && \text{(Theorem 5.4.1)} \\ &= i_{\mathsf{A}^{\mathcal{V}_{\underline{\mathsf{R}}}}}(t^{\mathcal{V}_{\underline{\mathsf{R}}}\mathcal{V}_{\underline{\mathsf{R}}}}[i^{-1}(\Gamma^{\mathcal{V}_{\underline{\mathsf{R}}}})]) && \text{(Definition of } u\text{)} \\ &= t && \text{(Theorem 5.4.1)} \ . \end{aligned}$$

The proofs of statements 3 and 4 are similar. □

6. Recovering linear-use CPS translations of typed lambda-calculus

In this section, we use the self-translation to establish properties of the call-by-value and call-by-name linear-use CPS translations of Section 4. The main property we exploit is that the generic self-translation subsumes the call-by-value and call-by-name translations. Indeed, the latter are obtained uniformly by precomposing the generic self-translation on EEC with the standard call-by-value and call-by-name translations from λ-calculus to EEC, given in Section 3.

Theorem 6.1 (Recovering $(\cdot)^{\mathrm{v}_{\underline{\mathsf{R}}}}$)**.** *For every simple type* σ*, we have* $\sigma^{\mathrm{v}_{\underline{\mathsf{R}}}} = (\sigma^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}$*; and, for every simply-typed term* $\Theta \vdash M\colon \sigma$*, we have* $\Theta^{\mathrm{v}_{\underline{\mathsf{R}}}} \mid\!- \vdash M^{\mathrm{v}_{\underline{\mathsf{R}}}} = (M^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}\colon (\sigma^{\mathrm{v}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}$*.*

Theorem 6.2 (Recovering $(\cdot)^{\mathrm{n}_{\underline{\mathsf{R}}}}$)**.** *Suppose* $\underline{\mathsf{R}}$ *is different from* $\underline{\alpha}$*, for every simply-typed* λ*-calculus type constant* α*. Then, for every* σ*, we have* $\sigma^{\mathrm{n}_{\underline{\mathsf{R}}}} = (\sigma^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}}$*, hence* $\sigma^{\mathrm{n}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}} = (\sigma^{\mathrm{n}})^{\mathcal{V}_{\underline{\mathsf{R}}}}$*; and, for every term* $\Theta \vdash M\colon \sigma$*, we have* $\Theta^{\mathrm{n}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}} \mid\!- \vdash M^{\mathrm{n}_{\underline{\mathsf{R}}}} = (M^{\mathrm{n}})^{\mathcal{V}_{\underline{\mathsf{R}}}}\colon \sigma^{\mathrm{n}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}}$*.*

The proofs are by induction on the structure of σ and M.

Proof of Theorem 6.1. For the type equality, we consider the case of $\sigma \to \tau$.

$$\begin{aligned}
((\sigma \to \tau)^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} &= (\sigma^{\mathrm{v}} \to {!}\,\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
&= (\sigma^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \to ({!}\,\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
&= (\sigma^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \to (({!}\,\tau^{\mathrm{v}})^{\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}}) \\
&= (\sigma^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \to (((\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}) \\
&= \sigma^{\mathrm{v}_{\underline{\mathsf{R}}}} \to ((\tau^{\mathrm{v}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}) \qquad \text{by induction hypothesis} \\
&= (\sigma \to \tau)^{\mathrm{v}_{\underline{\mathsf{R}}}} \ .
\end{aligned}$$

And, in the case of the term MN, where $\Theta \vdash M \colon \sigma \to \tau$ and $\Theta \vdash N \colon \sigma$, we have:

$$\begin{aligned}
((MN)^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} &= (\mathrm{let}\ {!}f\ \mathrm{be}\ M^{\mathrm{v}}\ \mathrm{in}\ \mathrm{let}\ {!}x\ \mathrm{be}\ N^{\mathrm{v}}\ \mathrm{in}\ f(x))^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
&= \lambda^{\circ} k \colon (\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}.\ (M^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[\underline{\lambda} f \colon ((\sigma \to \tau)^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ (\mathrm{let}\ {!}x\ \mathrm{be}\ N^{\mathrm{v}}\ \mathrm{in}\ f(x))^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[k]] \\
&= \lambda^{\circ} k \colon (\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}.\ (M^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[\underline{\lambda} f \colon ((\sigma \to \tau)^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ (N^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[\underline{\lambda} x \colon (\sigma^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathsf{R}}}}.\ f(x)[k]]] \\
&= \lambda^{\circ} k \colon \tau^{\mathrm{v}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}.\ M^{\mathrm{v}_{\underline{\mathsf{R}}}}\,[\underline{\lambda} f \colon \sigma^{\mathrm{v}_{\underline{\mathsf{R}}}} \to (\tau^{\mathrm{v}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}.\ N^{\mathrm{v}_{\underline{\mathsf{R}}}}\,[\underline{\lambda} x \colon \sigma^{\mathrm{v}_{\underline{\mathsf{R}}}}.\ f(x)[k]]] \\
&= (MN)^{\mathrm{v}_{\underline{\mathsf{R}}}} \ .
\end{aligned}$$

□

Proof of Theorem 6.2. First, we observe that for a type-constant α, we have

$$\alpha^{\mathrm{n}_{\underline{\mathsf{R}}}} = \underline{\alpha} = \underline{\alpha}^{\mathcal{C}_{\underline{\mathsf{R}}}} = (\alpha^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}} \ ,$$

where the middle equality relies on the asumption that $\underline{\alpha}$ is different from $\underline{\mathsf{R}}$.

Of the other cases, we again consider $\sigma \to \tau$.

$$\begin{aligned}
((\sigma \to \tau)^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}} &= (\sigma^{\mathrm{n}} \Rightarrow \tau^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}} \\
&= {!}((\sigma^{\mathrm{n}})^{\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes (\tau^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}} \\
&= {!}((\sigma^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}}) \otimes (\tau^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}} \\
&= {!}(\sigma^{\mathrm{n}_{\underline{\mathsf{R}}}} \multimap \underline{\mathsf{R}}) \otimes \tau^{\mathrm{n}_{\underline{\mathsf{R}}}} \qquad \text{by induction hypothesis} \\
&= (\sigma \to \tau)^{\mathrm{n}_{\underline{\mathsf{R}}}} \ .
\end{aligned}$$

And the case of an application MN works out as:

$$\begin{aligned}
((MN)^{\mathrm{n}})^{\mathcal{V}_{\underline{\mathsf{R}}}} &= (M^{\mathrm{n}}\,\underline{(N^{\mathrm{n}})})^{\mathcal{V}_{\underline{\mathsf{R}}}} \\
&= \lambda^{\circ} k \colon (\tau^{\mathrm{n}})^{\mathcal{C}_{\underline{\mathsf{R}}}}.\ (M^{\mathrm{n}})^{\mathcal{V}_{\underline{\mathsf{R}}}}\,[{!}((N^{\mathrm{n}})^{\mathcal{V}_{\underline{\mathsf{R}}}}) \otimes k] \\
&= \lambda^{\circ} k \colon \tau^{\mathrm{n}_{\underline{\mathsf{R}}}}.\ M^{\mathrm{n}_{\underline{\mathsf{R}}}}\,[{!}(N^{\mathrm{n}_{\underline{\mathsf{R}}}}) \otimes k] \\
&= (MN)^{\mathrm{n}_{\underline{\mathsf{R}}}} \ .
\end{aligned}$$

□

We comment that many of the syntactic choices of this paper have been made in order to obtain Theorems 6.1 and 6.2 in the simple form stated. For example, in the conference version of the paper [EMS10], where neither value-type and computation-type products nor value- and computation-type function spaces are distinguished syntactically, Theorem 6.2 holds only up to type isomorphism, rather than up to equality. Similarly, had a different choice been made for $(\sigma \to \tau)^{\mathrm{v}_{\underline{\mathsf{R}}}}$ in Figure 5, for example $(\tau^{\mathrm{v}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}}) \multimap (\sigma^{\mathrm{v}_{\underline{\mathsf{R}}}} \Rightarrow \underline{\mathsf{R}})$, as discussed in Section 4, then Theorem 6.1 would have held only up to isomorphism.

Using Theorems 6.1 and 6.2, it is now straightforward to provide the postponed proofs of soundness and full completeness for the cbv and cbn linear-use CPS translations of simply-typed λ-calculus from Section 4, by deriving these results as consequences of soundness and full completeness for the self-translation (Theorems 5.2 and 5.5). We give the proofs for the call-by-value case only (Proposition 4.1 and Theorem 4.3). The proofs for the corresponding call-by-name results (Proposition 4.2 and Theorem 4.4), are similarly straightforward.

Proof of Proposition 4.1 (Soundness $(\cdot)^{\mathrm{v}\underline{\mathrm{R}}}$). Suppose $\Theta \vdash M =_{\lambda_c} N \colon \tau$. Proposition 3.1.1 shows $\Theta^{\mathrm{v}} \mid - \vdash M^{\mathrm{v}} = N^{\mathrm{v}} \colon !(\tau^{\mathrm{v}})$. Whence, by Theorem 5.2, $(\Theta^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \mid - \vdash (M^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} = (N^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \colon (!\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}}$. That is, by Theorem 6.1, $\Theta^{\mathrm{v}\underline{\mathrm{R}}} \mid - \vdash M^{\mathrm{v}\underline{\mathrm{R}}} = N^{\mathrm{v}\underline{\mathrm{R}}} \colon (\tau^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathrm{R}}) \multimap \underline{\mathrm{R}}$. □

Proof of Theorem 4.3 (Full completeness of $(\cdot)^{\mathrm{v}\underline{\mathrm{R}}}$). For statement 1, suppose $\Theta \vdash M, N \colon \tau$ and $\Theta^{\mathrm{v}\underline{\mathrm{R}}} \mid - \vdash M^{\mathrm{v}\underline{\mathrm{R}}} = N^{\mathrm{v}\underline{\mathrm{R}}} \colon (\tau^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathrm{R}}) \multimap \underline{\mathrm{R}}$. By Theorem 6.1, this is equivalent to $(\Theta^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \mid - \vdash (M^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} = (N^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \colon (!\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}}$. So, by Theorem 5.5.1, $\Theta^{\mathrm{v}} \mid - \vdash M^{\mathrm{v}} = N^{\mathrm{v}} \colon !(\tau^{\mathrm{v}})$. Whence, by Proposition 3.1.2, $\Theta \vdash M =_{\lambda_c} N \colon \tau$, as required.

For statement 2, suppose $\Theta^{\mathrm{v}\underline{\mathrm{R}}} \mid - \vdash t \colon (\tau^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathrm{R}}) \multimap \underline{\mathrm{R}}$. That is, by Theorem 6.1, $(\Theta^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \mid - \vdash t \colon (!\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}}$. Then, by Theorem 5.5.2 there exists $\Theta^{\mathrm{v}} \mid - \vdash u \colon !\tau^{\mathrm{v}}$ such that $(\Theta^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \mid - \vdash t = u^{\mathcal{V}_{\underline{\mathrm{R}}}} \colon (!\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}}$. And, by Proposition 3.1.3, there exists $\Theta \vdash M \colon \tau$ such that $\Theta^{\mathrm{v}} \mid - \vdash u = M^{\mathrm{v}} \colon !\tau^{\mathrm{v}}$. Therefore, by Theorem 5.2, $(\Theta^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \mid - \vdash t = (M^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}} \colon (!\tau^{\mathrm{v}})^{\mathcal{V}_{\underline{\mathrm{R}}}}$. That is, again by Theorem 6.1, $\Theta^{\mathrm{v}\underline{\mathrm{R}}} \mid - \vdash t = M^{\mathrm{v}\underline{\mathrm{R}}} \colon (\tau^{\mathrm{v}\underline{\mathrm{R}}} \Rightarrow \underline{\mathrm{R}}) \multimap \underline{\mathrm{R}}$, as required. □

7. Perspectives

Throughout the paper, we have taken EEC for granted. However, linear-use CPS translations can themselves be used as a motivation for the selection of type constructors appearing in EEC. Given Hasegawa's call-by-value and call-by-name linear-use CPS translations into ILL [Has02, Has04], it is natural to ask if these translations can be encompassed within a single linear-use CPS translation of Levy's CBPV into ILL — since one of the *raisons d'être* of CBPV is to have a uniform language generalising cbv and cbn [Lev04]. For our *effect calculus* (EC), that is, for CBPV without value-type sums (see the discussion in Section 2), the answer is provided by our generic self-translation on EEC. A linear-use CPS translation of the effect calculus is obtained by restricting the source of the self-translation to EC, and by reinterpreting the target of the translation as ILL. Having done this, one sees that the fragment of ILL that is used in performing this translation is EEC. Thus EEC arises as naturally the smallest fragment of ILL able to act as a target language for a linear-use CPS translation of EC. Value-type sums, that is the whole of CBPV, can be accommodated in the picture by simply adding value-type sums to EEC, see [EMS12]. The generic self-translation of Section 5 easily extends to a self-translation on the resulting system EEC+. Thus there is a linear-use CPS translation of full CBPV into EEC+.[6]

It is a remarkable fact that EEC supports its own linear-use CPS translation as a self-translation. As we have seen, this property does not hold of smaller fragments, such as the effect calculus, whose linear-use CPS translation requires the full expressivity of EEC. It also does not extend to ILL itself. That is, the linear-use CPS translation of EEC cannot be extended to obtain an analogous linear-use CPS translation from ILL to itself. To appreciate

[6]It is less straightforward to give a linear-use CPS translation of the whole of CBPV into ILL. Because there is no distinction between "linear" and "intuitionistic" types, analogous to the distinction between computation and value types, there is no natural interpretation for value-type sums in ILL. Sums are best incorporated by moving to a version of linear logic that includes such a type distinction [Ben95].

this, it is necessary to say something about the category-theoretic model theory underlying linear-use CPS translations. This model theory provides an illuminating perspective on the syntactic material presented in the paper.

Roughly speaking, a *model* $\mathcal{M}$ of EEC is given by a tuple

$$(\mathcal{V},\, \mathcal{C},\, F \dashv G \colon \mathcal{C} \to \mathcal{V}) \ ,$$

where: $\mathcal{V}$ is a category modelling functions between value types; $\mathcal{C}$ is a category modelling linear functions between computation types; and $F \dashv G$ is an adjunction, with F modelling the $!(\cdot)$ type construction, and G providing the coercion from computation types to value types. A significant amount of additional structure, all of which is determined by universal properties, is also required on the categories, to interpret the other type constructors of EEC. The reader is referred to [EMS09, EMS12, EMS1x] for further details, which are somewhat technical — substantial use is made of enriched category theory [Kel82]. The point relevant to the content of the present paper is that the categorical models of EEC are closed under an interesting construction. Given a model as above, let $\underline{\mathsf{R}}$ be a chosen object of $\mathcal{C}$. We call the structure $(\mathcal{V},\, \mathcal{C},\, F \dashv G \colon \mathcal{C} \to \mathcal{V},\, \underline{\mathsf{R}})$ a *pointed model*. Such a pointed model, $\mathcal{N}$, has a *dual (pointed) model*:

$$\mathcal{N}^\star \;=\; (\mathcal{V},\, \mathcal{C}^{\mathrm{op}},\, F^\star \dashv G^\star \colon \mathcal{C}^{\mathrm{op}} \to \mathcal{V},\, \underline{\mathsf{I}}) \ ,$$

where $F^\star$ corresponds to the contravariant mapping $\mathsf{A} \mapsto \mathsf{A} \Rightarrow \underline{\mathsf{R}}$ from value to computation types, $G^\star$ corresponds to the contravariant mapping $\underline{\mathsf{A}} \mapsto \underline{\mathsf{A}} \multimap \underline{\mathsf{R}}$ in the other direction, and $\underline{\mathsf{I}}$ is the object of $\mathcal{C}$ chosen to model the type $\underline{\mathsf{I}}$ in $\mathcal{N}$. Thus the monad GF on $\mathcal{V}$, which models $!(\cdot)$ in $\mathcal{N}$, is converted to the monad $G^\star F^\star = ((\cdot) \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}$ on $\mathcal{V}$, which models $!(\cdot)$ in the dual model $\mathcal{N}^\star$. Monads of the form $G^\star F^\star$ have been called *dual monads* by Lawvere [Law69]. In our setting, the dual terminology is particularly apt, since we have:

Fact 7.1. Every pointed model $\mathcal{N}$ is isomorphic to its double dual $\mathcal{N}^{\star\star}$.

Importantly, the isomorphism preserves the pointed-model structure, but only up to coherent natural isomorphism. Up-to-isomorphism structure preservation is taken as the basic notion of morphism of EEC models [EMS09, EMS1x]. Those special morphisms that preserve structure up to equality ("on the nose") are referred to as *strict*.

Given the description of $G^\star F^\star$ as $((\cdot) \Rightarrow \underline{\mathsf{R}}) \multimap \underline{\mathsf{R}}$, a connection with linear-use CPS translations is apparent at the level of monads. Accordingly, one might call $\mathcal{N}^\star$ a linearly-used continuations model relative to $\mathcal{N}$. By Fact 7.1, every (pointed) model of EEC arises as a linearly-used continuations model relative to another model, namely relative to its own dual model — a property, which is somewhat surprising at first sight.

The dual monad construction also allows us to reconstruct the self-translation of Section 5 semantically. There is a syntactic model $\mathcal{M}_{\mathrm{syn}}$ whose objects are EEC types and whose morphisms are terms modulo provable equality. This enjoys an initiality property: for any interpretation of type constants in a model $\mathcal{M}$ there is a unique strict morphism of models from $\mathcal{M}_{\mathrm{syn}}$ to $\mathcal{M}$ that maps type constants in the specified way. Let $\underline{\mathsf{R}}$ be a chosen computation type. Define $\mathcal{N}_{\mathrm{syn}\underline{\mathsf{R}}}$ be the pointed model with $\underline{\mathsf{R}}$ as its point. Interpret all type constants as themselves, except for $\underline{\mathsf{R}}$ which, if it is a type constant, gets interpreted as $\underline{\mathsf{I}}$. Then the induced strict morphism of models from $\mathcal{M}_{\mathrm{syn}}$ to the underlying model of $\mathcal{N}^\star_{\mathrm{syn}\underline{\mathsf{R}}}$ is exactly the generic self-translation of Section 5. That is, the action of the morphism on (objects and morphisms of) $\mathcal{V}$ is given by $(\cdot)^{\mathcal{V}_{\underline{\mathsf{R}}}}$ (on types and terms respectively), and its action on $\mathcal{C}$ is given by $(\cdot)^{\mathcal{C}_{\underline{\mathsf{R}}}}$.

It is now possible to substantiate the claim made earlier that the self-translation of EEC does not extend to the whole of ILL. Any model of ILL (of the general form described in [Ben95]) is also a model of ECC. Let $\mathcal{N}$ be a pointed model of ILL. Then, in general, its dual $\mathcal{N}^\star$, although still a model of EEC, is not a model of ILL (the linear category need not be symmetric monoidal closed). In particular, when $\mathcal{N}$ is the syntactic (initial) model of ILL (with chosen $\underline{\mathsf{R}}$), the dual model $\mathcal{N}^\star$ is not a model of ILL. Thus there is no induced morphism of models from the syntactic ILL model to its dual. That is, there is no linear-use CPS translation of ILL to itself.

Returning to the self-translation of EEC, we now outline how the semantic perspective provides a conceptually clean proof of the involution property and full completeness. Suppose $\underline{\mathsf{R}}$ is either a type constant or $\underline{\mathsf{I}}$. Then the morphism from $\mathcal{M}_{\mathrm{syn}}$ to $\mathcal{N}^\star_{\mathrm{syn}\underline{\mathsf{R}}}$, described above as corresponding to the self-translation, extends (trivially) to a morphism of pointed models from $\mathcal{N}_{\mathrm{syn}\underline{\mathsf{R}}}$ to $\mathcal{N}^\star_{\mathrm{syn}\underline{\mathsf{R}}}$. The operation of taking duals is functorial (in an appropriate 2-categorical sense), and so we obtain a morphism of pointed models from $\mathcal{N}^\star_{\mathrm{syn}\underline{\mathsf{R}}}$ to $\mathcal{N}^{\star\star}_{\mathrm{syn}\underline{\mathsf{R}}}$; whence, by composition, a morphism from $\mathcal{N}_{\mathrm{syn}\underline{\mathsf{R}}}$ to $\mathcal{N}^{\star\star}_{\mathrm{syn}\underline{\mathsf{R}}}$. The composite morphism preserves type constants. Furthermore $\mathcal{M}_{\mathrm{syn}}$ enjoys a universal property with respect to non-strict morphisms: for any interpretation of type constants in a model $\mathcal{M}$ there is a unique-up-to-coherent-natural-isomorphism morphism of models from $\mathcal{M}_{\mathrm{syn}}$ to $\mathcal{M}$ that maps type constants (up to isomorphism) in the specified way. This means that, the induced morphism from $\mathcal{N}_{\mathrm{syn}\underline{\mathsf{R}}}$ to $\mathcal{N}^{\star\star}_{\mathrm{syn}\underline{\mathsf{R}}}$ is coherently naturally isomorphic to the morphism implementing the double-duality of Fact 7.1. This is literally the involution property of the self-translation (Theorem 5.4) in semantic form. With a little more manipulation of the universal property of $\mathcal{M}_{\mathrm{syn}}$, one obtains:

Fact 7.2. The morphism from $\mathcal{N}_{\mathrm{syn}\underline{\mathsf{R}}}$ to $\mathcal{N}^\star_{\mathrm{syn}\underline{\mathsf{R}}}$ is an equivalence of pointed models.

This result corresponds to the full completeness of the self translation (Theorem 5.5). Semantically, it states the surprising, at first sight, fact that the syntactic (pointed) model is self-dual.

There is, however, an alternative perspective on models, from which the self-duality of the initial model is less surprising. It is possible to omit the adjunction $F \dashv G$ from the structure of the model, and instead simply specify the object $\underline{\mathsf{I}}$. The adjunction is then recovered using the requirement that $\mathcal{C}$ have *copowers* (a concept from enriched category theory), which is part of the assumed structure of a model. The operation of taking the dual of a pointed model, with point $\underline{\mathsf{R}}$, then has a simple description: instead of redefining the adjunction, the rôles of the objects $\underline{\mathsf{I}}$ and $\underline{\mathsf{R}}$ are simply swapped in the structure.

Detailed definitions and proofs of all the semantic facts referred to above in this section will appear in a paper devoted entirely to the category-theoretic models of EEC [EMS1x]. Unfortunately, although the high-level ideas are straightforward, considerable technicalities arise in getting the details correct. The reader who wishes to see a slightly fuller treatment than the outline given above, but not all details, is referred to the conference version of the present paper [EMS10].

To finish, we return to syntax. The alternative formulation of models, referred to above, has a syntactic counterpart. Since $\underline{\mathsf{I}}$ is included as a primitive computation type in our formulation of EEC, it would be possible to omit, from the syntax of EEC, both the type constructor $!\mathsf{A}$ and the inclusion of computation types amongst value types. The former can be defined as $!\mathsf{A} \otimes \underline{\mathsf{I}}$. And the value type corresponding to a computation type $\underline{\mathsf{A}}$ can be recovered as $\underline{\mathsf{I}} \multimap \underline{\mathsf{A}}$. (Thus Levy's U constructor [Lev04] is rendered visible.)

Using this restricted syntax, the involution property of Theorem 5.4 has a simplified form. There is no longer any need for the isomorphisms i_{A} and $j_{\underline{\mathsf{A}}}$, since one obtains identities $\mathsf{A}^{\mathcal{V}_{\mathrm{R}}\mathcal{V}_{\mathrm{R}}} = \mathsf{A}$ and $\underline{\mathsf{A}}^{\mathcal{C}_{\mathrm{R}}\mathcal{C}_{\mathrm{R}}} = \underline{\mathsf{A}}$. The very mild drawback of this formulation is that it requires the slightly more complex definition of $(\sigma \to \tau)^{\mathrm{v}} = \sigma^{\mathrm{v}} \to (\underline{\mathsf{I}} \multimap !\tau^{\mathrm{v}} \otimes \underline{\mathsf{I}})$ in Figure 4. Or alternatively, one could take $(\sigma \to \tau)^{\mathrm{v}} = !\sigma^{\mathrm{v}} \otimes \underline{\mathsf{I}} \multimap !\tau^{\mathrm{v}} \otimes \underline{\mathsf{I}}$, which would fit in with redefining $(\sigma \to \tau)^{\mathrm{v}_{\mathrm{R}}} = (\tau^{\mathrm{v}_{\mathrm{R}}} \Rightarrow \underline{\mathsf{R}}) \multimap (\sigma^{\mathrm{v}_{\mathrm{R}}} \Rightarrow \underline{\mathsf{R}})$, as discussed in Section 4, leaving the value-type-function-space constructor, $\to$, superfluous to the translations.

However, for the present paper, we have preferred to retain $!(\cdot)$ as a primitive type construct, due to the basic rôle it plays in related type systems: as T in Moggi's computational metalanguage [Mog91], as F in Levy's CBPV [Lev04], and as the exponential in linear logic [Gir87]. For one thing, our choice of primitives has allowed us to give the various translations of Sections 3 and 4 just as they appear in the literature [Mog91, Fil96, Lev04, Has02, Has04], modulo the change to EEC notation. We also comment that it is perhaps the standard focus on $!(\cdot)$ (or T or F) as the key construct in effect languages that makes the involution property of the self-translation translation come as a surprise when first encountered. For, amongst computation types, the type $!\mathsf{A}$ has the most interesting translation — the only one which is not part of a dual pair. It is for this reason that the proofs of Section 5 mainly focus on constructs associated with types of the form $!\mathsf{A}$ as providing the interesting cases.

In the present paper, we have investigated the enriched effect calculus as a metalanguage for formalising one possible interaction between linearity and CPS translations. It is the belief of the authors that EEC will prove a useful language for modelling other ways in which linearity and effects combine. Some potential examples of such interactions are briefly discussed in the main paper introducing EEC [EMS12]. It would be interesting to see further convincing examples worked out in detail.

Acknowledgements

We thank Masahito Hasegawa, Paul Levy and the anonymous referees for helpful suggestions.

References

[Bar97] A. Barber. *Linear Type Theories, Semantics and Action Calculi*. PhD thesis, Department of Computer Science, University of Edinburgh, 1997.

[Ben95] P. N. Benton. A mixed linear and non-linear logic: Proofs, terms and models. In *Proc. Computer Science Logic (CSL) 1994*, volume 933 of *LNCS*. Springer, 1995.

[BORT02] J. Berdine, P. W. O'Hearn, U. Reddy, and H. Thielecke. Linear continuation-passing. *Higher Order and Symbolic Computation*, 15:181–208, 2002.

[BW96] P. N. Benton and P. Wadler. Linear logic, monads, and the lambda calculus. In *Proc. 11th Annual Symposium on Logic in Computer Science (LICS)*, 1996.

[EMS09] J. Egger, R. E. Møgelberg, and A. Simpson. Enriching an effect calculus with linear types. In *Proc. Computer Science Logic (CSL)*, volume 5771 of *LNCS*, pages 240–254. Springer, 2009.

[EMS10] J. Egger, R. E. Møgelberg, and A. Simpson. Linearly-used continuations in the enriched effect calculus. In *Proc. Foundations of Software Science and Computation Structures (FoSSaCS)*, volume 6014 of *LNCS*, pages 18–32. Springer, 2010.

[EMS12] J. Egger, R. E. Møgelberg, and A. Simpson. The enriched effect calculus: Syntax and semantics. *Journal of Logic and Computation*, Advance Access published June 19, 2012. doi: 10.1093/logcom/exs025.

[EMS1x] J. Egger, R. E. Møgelberg, and A. Simpson. Categorical models for the enriched effect calculus, 201x. In preparation.

[Fil96] A. Filinski. *Controlling Effects*. PhD thesis, Carnegie Mellon University, 1996.

[Gir87] J.-Y. Girard. Linear logic. *Theoretical Computer Science*, 50:1–102, 1987.

[Has02] M. Hasegawa. Linearly used effects: Monadic and CPS transformations into the linear lambda calculus. In *Proc. 6th International Symposium on Functional and Logic Programming (FLOPS)*, volume 2441 of *LNCS*, pages 167–182. Springer, 2002.

[Has04] M. Hasegawa. Semantics of linear continuation-passing in call-by-name. In *Proc. 7th International Symposium on Functional and Logic Programming (FLOPS)*, volume 2998 of *LNCS*, pages 229–243. Springer, 2004.

[Kel82] G. M. Kelly. *Basic Concepts of Enriched Category Theory*, volume 64 of *LMS Lecture Notes*. Cambridge University Press, 1982.

[Law69] F. W. Lawvere. Ordinal sums and equational doctrines. In *Seminar on Triples and Categorical Homology Theory (ETH, Zürich)*, pages 141–155. Springer, 1969.

[Lev04] P. B. Levy. *Call-by-push-value. A functional/imperative synthesis*. Semantic Structures in Computation. Springer, 2004.

[Lev05] P. B. Levy. Adjunction models for call-by-push-value with stacks. *Theory and Applications of Categories*, 14:75–110, 2005.

[Mog89] E. Moggi. Computational lambda-calculus and monads. In *Proc. 4th Annual Symposium on Logic in Computer Science (LICS)*, pages 14–23, 1989.

[Mog91] E. Moggi. Notions of computation and monads. *Information and Computation*, 93:55–92, 1991.

[Par92] M. Parigot. $\lambda\mu$-calculus: an algorithmic interpretation of classical natural deduction. In *Proc. Logic Programming and Automated Reasoning (LPAR)*, volume 624 of *LNCS*, pages 190–201. Springer, 1992.

[Plo75] G. D. Plotkin. Call-by-name, call-by-value and the lambda-calculus. *Theoretical Computer Science*, 1:125–159, 1975.

[RS98] B. Reus and Th. Streicher. Classical logic, continuation semantics and abstract machines. *Journal of Functional Programming*, 8:543–572, 1998.